



การตรวจจับและการจำแนกใบหน้าสวมหน้ากากโดยใช้ข้อมูลเข้าหลายค่า
กับการเรียนรู้เชิงลึก
Face-mask Detection and Classification Using Multiple Inputs
Deep Learning

ภิรมกานต์ รุจิราพงศ์
Peemakarn Rujirapong

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Science
Prince of Songkla University

2566

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์



การตรวจจับและการจำแนกใบหน้าสวมหน้ากากโดยใช้ข้อมูลเข้าหลายค่า
กับการเรียนรู้เชิงลึก

Face-mask Detection and Classification Using Multiple Inputs
Deep Learning

ภิรมกานต์ รุจิราพงศ์

Peemakarn Rujirapong

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา

วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Science

Prince of Songkla University

2566

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

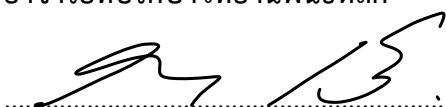
ชื่อวิทยานิพนธ์ การตรวจจับและการจำแนกใบหน้าสวมหน้ากากโดยใช้ข้อมูลเข้าหลายค่ากับ

การเรียนรู้เชิงลึก

ผู้เขียน นายภิมกานต์ รุจิราพงศ์

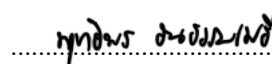
สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

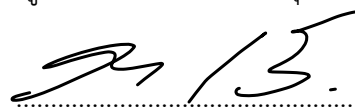


(ผู้ช่วยศาสตราจารย์ ดร.วิภาดา เวทย์ประสิทธิ์)


คณะกรรมการสอบ

.....ประธานกรรมการ


(ผู้ช่วยศาสตราจารย์ ดร.พุทธิพร ชนธรรมเมธี)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วิภาดา เวทย์ประสิทธิ์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ชินพงศ์ อังสุโชติเมธี)

.....กรรมการ

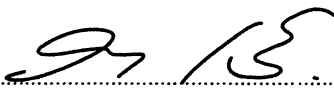
(ผู้ช่วยศาสตราจารย์ ดร.ภัทร อัยรักษ์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้รับวิทยานิพนธ์ฉบับนี้เป็น
ส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการ
คอมพิวเตอร์


.....
(ผู้ช่วยศาสตราจารย์ ดร.กวินพัฒน์ สิริกานติโสภณ)

รักษาการแทนคณบดีบัณฑิตวิทยาลัย

ขอรับรองว่า ผลงานวิจัยนี้มาจากการศึกษาวิจัยของนักศึกษาเอง และได้แสดงความขอบคุณบุคคลที่มีส่วนช่วยเหลือแล้ว

ลงชื่อ..... 

(ผู้ช่วยศาสตราจารย์ ดร.วิภาดา เวทย์ประสิทธิ์)
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ลงชื่อ..... 

(นายกิมกานต์ รุจิราพงศ์)
นักศึกษา

ข้าพเจ้าขอรับรองว่า ผลงานวิจัยนี้ไม่เคยเป็นส่วนหนึ่งในการอนุมัติปริญญาในระดับใดมาก่อน และ
ไม่ได้ถูกใช้ในการยื่นขออนุมัติปริญญาในขณะนี้

ลงชื่อ.....^{ผู้พิมพ์}.....

(นายภิรมกานต์ รุจิราพงศ์)

นักศึกษา

ชื่อวิทยานิพนธ์	การตรวจจับและการจำแนกใบหน้าสวมหน้ากากโดยใช้ข้อมูลเข้าหลายค่ากับการเรียนรู้เชิงลึก
ผู้เขียน	นายภิมกานต์ รุจิราพงศ์
สาขาวิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2566

บทคัดย่อ

งานวิจัยนี้นำเสนอการตรวจจับและการจำแนกใบหน้าสวมหน้ากากโดยใช้ข้อมูลเข้าหลายค่า เนื่องจากการตรวจจับใบหน้าสวมหน้ากากแบบ 3 ประเภท ได้แก่ ประเภทสวมหน้ากากถูกต้อง ประเภทไม่สวมหน้ากาก และประเภทสวมหน้ากากไม่ถูกต้อง เมื่อสังเกตข้อแตกต่างของทั้ง 3 ประเภท จะเห็นว่าบริเวณจมูกและปากจะเป็นส่วนหลักที่ช่วยในการจำแนกประเภท จึงพัฒนาโมเดลการเรียนรู้เชิงลึกที่รับข้อมูลเข้าหลายค่า ได้แก่ ภาพใบหน้า ภาพบริเวณจมูกและภาพบริเวณปาก การทดลองมีการใช้ชุดข้อมูล 2 ชุด ได้แก่ ชุดข้อมูล Face Mask Label Dataset (FMLD) และ Andrewmvd Face Mask Detection Kaggle (AFMDK) มีการเปรียบเทียบการสร้างโมเดลซึ่งใช้ข้อมูลเข้าแบบเดี่ยว พบว่าผลลัพธ์ที่ได้โมเดลข้อมูลเข้าหลายค่าที่นำเสนอมีค่าความถูกต้อง ค่าความแม่นยำ ค่าความครบถ้วน และค่า F1 มีประสิทธิภาพสูงกว่าโมเดลข้อมูลเข้าเดี่ยวทั้งสองชุดข้อมูล นอกจากนี้งานวิจัยนี้ได้ทำการทดลองการปรับปรุงภาพ โดยใช้ Super-Resolution (SR) เพื่อแก้ปัญหาภาพขนาดเล็ก ทำให้ภาพบริเวณจมูกและภาพส่วนปากมีความคมชัดมากขึ้น จากการทดลองพบว่าโมเดลใช้ภาพจาก BSRGANs ไม่สามารถช่วยแก้ปัญหาภาพขนาดเล็กได้ แต่สามารถช่วยแก้ปัญหาภาพขนาดปานกลางและขนาดใหญ่ได้

คำสำคัญ : การจำแนกประเภทใบหน้า, โครงข่ายประสาทเทียมแบบคอนโวลูชัน, การสกัดคุณลักษณะ, การเรียนรู้เชิงลึก

Thesis Title	Face-mask Detection and Classification Using Multiple Inputs Deep Learning
Author	Mr. Peemakarn Rujirapong
Major Program	Computer Science
Academic Year	2023

ABSTRACT

This research presents face-mask detection and classification using multiple inputs. The face-mask detection composes of 3 classes including the with_mask class, with_out_mask class, and wear_mask_incorrect class. The differences between these classes are the nose area and mouth area which help in classification. A deep learning multiple input model that can use face images, nose images, and mouth images as inputs was developed. This experiment is tried out with 2 datasets including Face Mask Label Dataset (FMLD) and Andrewmvd Face Mask Detection Kaggle (AFMDK). There are comparison models which are created by using single input and multiple inputs. The study finds that the results are confirmed that the purposed multiple input model has accuracy, precision, recall, and F1 score has higher values than a single input model in both datasets. This research also does an experiment on image enhancement by super-resolution for small image problems. The results increase the resolution at the nose and mouth area. The experiment shows that model trained by image from BSRGANs cannot solves the small image problem but can solve the medium and large images.

Keyword : Face Classification, Convolutional Neural Network, Feature Extraction, Deep Learning

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สามารถสำเร็จลุล่วงได้ด้วยความช่วยเหลือและการสนับสนุนจากบุคคลหลายท่านที่คอยชี้แนะ และช่วยเหลือในด้านต่าง ๆ ผู้จัดทำวิทยานิพนธ์ขอขอบคุณทุกท่านเป็นอย่างมาก ณ โอกาสนี้

ขอขอบคุณอาจารย์ที่ปรึกษาวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. วิภาดา เวทย์ประสิทธิ์ และผู้ร่วมวิจัย ดร. นิวรรณ วัฒนกิจรุ่งโรจน์ ที่สละเวลา คอยชี้แนะ ให้คำแนะนำ และตรวจทานงานให้แก่ผู้ทำวิทยานิพนธ์ ตลอดจนวิทยานิพนธ์แล้วเสร็จ

คณาจารย์สาขาวิชาวิทยาการคอมพิวเตอร์ สาขาวิทยาศาสตร์การคำนวณ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ทุกท่านที่ให้ความรู้ด้านวิชาการและให้คำปรึกษา ตลอดจนทั้งคำแนะนำ ตอรายงานความก้าวหน้า

สุดท้ายนี้ขอขอบพระคุณบิดามารดา และเพื่อน ๆ พี่ ๆ ทุกคนที่คอยช่วยเหลือ นำมาซึ่งความสำเร็จในครั้งนี้

ภิรมกานต์ รุจิราพงศ์

สารบัญ

สารบัญ.....	(8)
รายการตาราง	(11)
รายการภาพประกอบ	(14)
บทที่ 1 บทนำ	1
1.1. ความสำคัญและที่มาของงานวิจัย	1
1.2. วัตถุประสงค์.....	2
1.3. ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4. ขอบเขตงานวิจัย.....	2
1.5. ขั้นตอนและระยะเวลาการดำเนินการ.....	2
1.5.1. ระยะเวลาการดำเนินการ	2
1.5.2. แผนการดำเนินการ	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1. การตรวจจับวัตถุ (Object Detection).....	4
2.2. โครงข่ายประสาทเทียม (Artificial Neural Networks : ANN).....	7
2.3. โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks (CNN))	8
2.4. โมเดลการเรียนรู้เชิงลึกแบบ ResNet	10
2.5. การเตรียมข้อมูลภาพก่อนการประมวลผล (Image Preprocessing).....	12
2.6. การปรับปรุงภาพ (Image Enhancement).....	13
2.7. การปรับปรุงภาพโดยใช้โมเดล CodeFormer และ BSRGANs.....	13
2.8. การสกัดคุณลักษณะ (Feature Extraction)	16
2.9. การจำแนกประเภท (Classification).....	17
2.10. การจำกัดตำแหน่ง (Localization).....	17
2.11. การตรวจจับใบหน้า (Face Detection).....	17

สารบัญ (ต่อ)

2.12. โมเดลการเรียนรู้เชิงลึกแบบ Multi-task Cascaded Convolutional Networks (MTCNN).....	18
2.13. การเรียนรู้แบบถ่ายโอน (Transfer Learning)	20
2.14. การวัดประสิทธิภาพของโมเดลการจำแนกประเภทข้อมูล	20
2.14.1. เทคนิคการแบ่งชุดข้อมูล และเทคนิคการประเมินประสิทธิภาพโมเดล	20
2.14.2. การประเมินประสิทธิภาพโมเดล.....	21
2.15. งานวิจัยที่เกี่ยวข้อง.....	26
บทที่ 3 วิธีการพัฒนาโมเดลข้อมูลเข้าหลายค่า	33
3.1. การออกแบบสถาปัตยกรรมและขั้นตอนวิธีสร้างและทดสอบโมเดล	33
3.2. การตรวจจับใบหน้า จมูก และปากโดยใช้ MTCNN (Face Nose Mouth Detection).....	45
3.3. การสกัดคุณลักษณะ (Feature Extraction) และการจำแนกประเภท (Classification).....	49
3.4. การปรับปรุงภาพโดยโมเดลการเรียนรู้เชิงลึกแบบ BSRGANs.....	53
บทที่ 4 การทดลองและผลการทดลอง	55
4.1. ชุดข้อมูลที่ใช้ในการทดลอง.....	55
4.2. ประสิทธิภาพการตรวจจับใบหน้าโดยใช้ MTCNN.....	61
4.3. การเตรียมชุดข้อมูลที่ใช้ในการทดลองโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่า	62
4.4. การกำหนดพารามิเตอร์การฝึกโมเดลและทดสอบโมเดลข้อมูลเข้าเดี่ยวและข้อมูลเข้าหลายค่า	66
4.5. การทดลองโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่า	66
4.5.1. เปรียบเทียบประสิทธิภาพของโมเดลที่ใช้ข้อมูลเข้าแบบเดี่ยวและข้อมูลเข้าหลายค่าบนชุดข้อมูล FMLD 5K.....	67

สารบัญ (ต่อ)

4.5.2. เปรียบเทียบประสิทธิภาพของโมเดลที่ใช้ข้อมูลเข้าแบบเดี่ยวและข้อมูลเข้าหลายค่าบนชุดข้อมูล FMLD MTCNN.....	70
4.5.3. เปรียบเทียบประสิทธิภาพของโมเดลที่ใช้ข้อมูลเข้าแบบเดี่ยวและข้อมูลเข้าหลายค่าบนชุดข้อมูล AFMDK.....	74
4.6. การเพิ่มประสิทธิภาพของโมเดลโดยใช้การปรับปรุงภาพ	76
4.6.1. การปรับปรุงภาพโดยใช้ Super-Resolution (SR)	77
4.6.2. เปรียบเทียบประสิทธิภาพของโมเดลเพื่อการปรับปรุงภาพ	85
บทที่ 5 บทสรุปและข้อเสนอแนะ	94
5.1. สรุปผลการวิจัย.....	94
5.2. ปัญหาและอุปสรรค	94
5.3. ข้อเสนอแนะ.....	95
เอกสารอ้างอิง	96
ภาคผนวก ก.....	104
ประวัติผู้เขียน.....	111

รายการตาราง

ตารางที่ 1 แผนการดำเนินงาน	3
ตารางที่ 2 ตาราง Confusion Matrix.....	21
ตารางที่ 3 แสดงรายละเอียดบทความเกี่ยวกับการตรวจจับใบหน้าสวมหน้ากากที่ได้ศึกษามา.....	6
ตารางที่ 4 การตรวจจับใบหน้าโดยใช้แนวคิดพื้นที่แสดงโมเดล และการตรวจจับบริเวณที่สนใจ	28
ตารางที่ 5 การตรวจจับใบหน้าโดยใช้แนวคิดเชิงถดถอยแสดงโมเดลต่าง ๆ ที่ใช้งาน.....	29
ตารางที่ 6 ผลลัพธ์ของบทความที่ใช้หน่วยวัดประสิทธิภาพเป็นค่าความถูกต้องของชุดข้อมูลต่าง ๆ.	31
ตารางที่ 7 ผลลัพธ์ของบทความที่ใช้หน่วยวัดประสิทธิภาพเป็น mAP ของชุดข้อมูลต่าง ๆ.....	32
ตารางที่ 8 แสดงจำนวนภาพใบหน้าแบ่งตามประเภทของภาพใบหน้าของชุดข้อมูล AFMDK.....	59
ตารางที่ 9 แสดงจำนวนภาพใบหน้าแบ่งตามประเภทของชุดข้อมูลและประเภทของภาพใบหน้าของ ชุดข้อมูล FMLD MTCNN	64
ตารางที่ 10 แสดงจำนวนภาพใบหน้าแบ่งตามประเภทของชุดข้อมูลและประเภทของภาพใบหน้าของ ชุดข้อมูล FMLD 5K.....	65
ตารางที่ 11 แสดงจำนวนภาพใบหน้าแบ่งตามประเภทของชุดข้อมูลและประเภทของภาพใบหน้าของ ชุดข้อมูล AFMDK.....	65
ตารางที่ 12 ผลการทดลองบนชุดข้อมูล FMLD 5K (2 class) ของโมเดลข้อมูลเข้าเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input)	68
ตารางที่ 13 ผลการทดลองบนชุดข้อมูล FMLD 5K (3 class) ของโมเดลข้อมูลเข้าเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input)	68
ตารางที่ 14 ผลการทดลองบนชุดข้อมูล FMLD MTCNN (3 class) ของโมเดลข้อมูลเข้าเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input).....	71
ตารางที่ 15 ผลการทดลองบนชุดข้อมูล FMLD MTCNN (3 class) ของโมเดลข้อมูลเข้าเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input).....	72
ตารางที่ 16 ผลการทดลองชุดข้อมูล AFMDK (2 class) โดยใช้โมเดลข้อมูลเข้าแบบเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input).....	74
ตารางที่ 17 ผลการทดลองชุดข้อมูล AFMDK (3 class) โดยใช้โมเดลข้อมูลเข้าแบบเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input).....	75

รายการตาราง (ต่อ)

ตารางที่ 18 แสดงรายละเอียดของชุดข้อมูล FMLD 5K แบ่งเป็นชุดข้อมูลทั้งหมด ชุดข้อมูลสำหรับฝึก และชุดข้อมูลสำหรับทดสอบ.....	76
ตารางที่ 19 แสดงรายละเอียดของชุดข้อมูล AFMDK แบ่งเป็นชุดข้อมูลทั้งหมด ชุดข้อมูลสำหรับฝึก และชุดข้อมูลสำหรับทดสอบ.....	77
ตารางที่ 20 ผลการทดลองโมเดลข้อมูลเข้าเดี่ยว (Single Input) ใช้ชุดข้อมูล FMLD 5K (3 class) โดยใช้โมเดลที่ฝึกโดยภาพ Non_SR และโมเดลที่ฝึกโดยภาพ SR	85
ตารางที่ 21 ผลการทดลองโมเดลข้อมูลเข้าหลายค่า (Multiple Input) ใช้ชุดข้อมูล FMLD 5K (3 class) โดยใช้โมเดลที่ฝึกโดยภาพ Non_SR และโมเดลที่ฝึกโดยภาพ SR	86
ตารางที่ 22 ผลการทดลองโมเดลข้อมูลเข้าหลายค่า (Multiple Input) ใช้ชุดข้อมูล FMLD 5K (3 class) ของโมเดลที่ฝึกโดยภาพ SR แบบแบ่งตามประเภท	86
ตารางที่ 23 ผลการทดลองโมเดลข้อมูลเข้าเดี่ยว (Single Input) ใช้ชุดข้อมูล AFMDK (3 class) โดยใช้โมเดลที่ฝึกโดยภาพ Non_SR และโมเดลที่ฝึกโดยภาพ SR	87
ตารางที่ 24 ผลการทดลองโมเดลข้อมูลเข้าหลายค่า (Multiple Input) ใช้ชุดข้อมูล AFMDK (3 class) โดยใช้โมเดลที่ฝึกโดยภาพ Non_SR และโมเดลที่ฝึกโดยภาพ SR	87
ตารางที่ 25 ผลการทดลองโมเดลข้อมูลเข้าหลายค่า (Multiple Input) ใช้ชุดข้อมูล AFMDK (3 class) ของโมเดลที่ฝึกโดยภาพ SR แบบแบ่งตามประเภท	87
ตารางที่ 26 ผลการเปรียบเทียบโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล FMLD 5K (3 class) ขนาดเล็กกว่า 50 พิกเซล ที่ฝึกและทดสอบโดยใช้ภาพ Non_SR และ SR.....	88
ตารางที่ 27 ผลการเปรียบเทียบโมเดลข้อมูลเข้าเดี่ยวที่ใช้ชุดข้อมูล AFMDK (3 class) ที่คัดเลือกภาพที่มีขนาดเล็กกว่า 50 พิกเซล ที่ฝึกและทดสอบโดยใช้ภาพ Non_SR และ SR	89
ตารางที่ 28 ผลการเปรียบเทียบโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล FMLD 5K (3 class) ขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล 50 พิกเซล ที่ฝึกและทดสอบโดยใช้ภาพ Non_SR และ SR	90
ตารางที่ 29 ผลการเปรียบเทียบโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล AFMDK (3 class) ที่คัดเลือกภาพที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล ที่ฝึกและทดสอบโดยใช้ภาพ Non_SR และ SR	91

รายการตาราง (ต่อ)

ตารางที่ 30 แสดงเวลาที่ใช้ในการทำงานของโมเดลตาม Algorithm1 (ไม่ปรับปรุงภาพ) และโมเดล
ตาม Algorithm2 (ปรับปรุงภาพ)..... 93

รายการภาพประกอบ

ภาพที่ 1 แสดงขั้นตอนการทำงานของ R-CNN	5
ภาพที่ 2 แสดงการทำงานของ YOLO	6
ภาพที่ 3 แสดงผลลัพธ์การทำงานของ	6
ภาพที่ 4 Neural Networks แบบ Fully-Connected Feedforward	7
ภาพที่ 5 การทำงานของนิวรอน	8
ภาพที่ 6 Convolutional Neural Networks	9
ภาพที่ 7 การทำงานของชั้น Convolutional.....	9
ภาพที่ 8 ภาพแสดง Identity Block (IB) Convolutional Block (CB) และ Layer ภายในของ Resnet.....	11
ภาพที่ 9 สถาปัตยกรรมของ Resnet152	12
ภาพที่ 10 ผลลัพธ์จากโมเดล CodeFormer.....	14
ภาพที่ 11 ผลลัพธ์ของ BSRGANs เมื่อเทียบกับโมเดลอื่น ๆ	15
ภาพที่ 12 ผลลัพธ์ของโมเดล BSRGANs บนภาพข้อมูลเข้า ล่างภาพข้อมูลออก	16
ภาพที่ 13 Haar Feature ที่หาข้อแตกต่างของค่าพิกเซลในพื้นที่สีขาวและสีดำ	18
ภาพที่ 14 ลำดับขั้นตอนการทำงานของ MTCNN	19
ภาพที่ 15 ความสัมพันธ์ระหว่างค่าความแม่นยำ ค่าความครบถ้วน และค่าความแม่นยำประมาณในช่วง	25
ภาพที่ 16 กราฟแสดงค่าความแม่นยำ ค่าความครบถ้วน และผลกระทบจากความแปรปรวน	25
ภาพที่ 17 สถาปัตยกรรมที่ออกแบบโมเดลการตรวจจับใบหน้าสวมหน้ากากข้อมูลเข้าหลายค่า	34
ภาพที่ 18 คำสั่งเพื่อสร้างตัวโหลดชุดข้อมูลที่ปรับค่าของพิกเซลให้อยู่ในช่วง 0-1	36
ภาพที่ 19 คำสั่งโหลดชุดข้อมูลสำหรับฝึก.....	36
ภาพที่ 20 คำสั่งโหลดข้อมูลสำหรับทดสอบ.....	36
ภาพที่ 21 คำสั่งสร้างตัวป้อนข้อมูลเข้าให้กับโมเดล โดยสามารถป้อนข้อมูลเข้าของภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก.....	37
ภาพที่ 22 สร้างตัวแปรเก็บข้อมูลเข้าสำหรับป้อนให้โมเดล	38
ภาพที่ 23 คำสั่งสำหรับสร้างเพื่อเตรียมข้อมูลออกของโมเดลข้อมูลเข้าหลายค่า	38

รายการภาพประกอบ (ต่อ)

ภาพที่ 24 สร้างชั้น Concatenate เพื่อรวมข้อมูลออกของโมเดลที่รับข้อมูลเข้าเป็นภาพใบหน้า ภาพ บริเวณจมูก และภาพบริเวณปาก.....	39
ภาพที่ 25 เชื่อมต่อชั้น Concatenate เข้ากับชั้น FC	39
ภาพที่ 26 สร้างส่วนรับข้อมูลเข้าที่สามารถรับข้อมูลเข้ามากกว่าหนึ่งค่า และสร้างโมเดลข้อมูลเข้า หลายค่า	40
ภาพที่ 27 คำสั่งเพื่อตั้งค่าโมเดลให้ใช้งาน optimizer ฟังก์ชันการสูญเสีย และตั้งค่าหน่วยวัด ประสิทธิภาพเป็นค่าความถูกต้อง	40
ภาพที่ 28 คำสั่งเพิ่มความสามารถในขณะฝึกให้ปรับค่าการเรียนรู้ลดลงเมื่อค่าความสูญเสียคงที่ หรือ หยุดการฝึก.....	40
ภาพที่ 29 คำสั่งที่ใช้สำหรับฝึกโมเดล	41
ภาพที่ 30 คำสั่งที่ใช้ทดสอบโมเดล	42
ภาพที่ 31 สรุปรูปโครงสร้างของโมเดลข้อมูลเข้าหลายค่า (โดยแสดงเฉพาะส่วนต้นและส่วนท้ายจาก ทั้งหมด 3,260 บรรทัด).....	43
ภาพที่ 32 ตัวอย่างภาพต้นฉบับที่มีภาพใบหน้ามากกว่าหนึ่งภาพ	46
ภาพที่ 33 ผลลัพธ์ของ MTCNN ระบุตำแหน่งของตา จมูก และปาก.....	46
ภาพที่ 34 ตัวอย่างภาพใบหน้า และผลลัพธ์จาก MTCNN บริเวณจมูกและปาก	47
ภาพที่ 35 โหลดชุดคำสั่งของ MTCNN	47
ภาพที่ 36 คำสั่งโหลดโมเดล MTCNN	47
ภาพที่ 37 คำสั่งใช้โมเดลตรวจจับใบหน้าจากภาพ	47
ภาพที่ 38 การหา IOU	48
ภาพที่ 39 กรอบภาพจริง กรอบภาพที่จำแนก และการใช้งาน IOU ใน Object Detection	49
ภาพที่ 40 การนำโมเดล Resnet 3 ตัวมาประกอบกันโดยใช้ชั้น Concatenation.....	49
ภาพที่ 41 ตัวอย่างการ Concatenation ของ Tensor.....	50
ภาพที่ 42 ตัวอย่างการ Concatenation ชุดตัวเลข.....	50
ภาพที่ 43 คำสั่งเพื่อสร้างส่วน Identity Block และ Convolutional Block ของ Resnet.....	51
ภาพที่ 44 คำสั่งสร้างโมเดล Resnet เริ่มต้นแต่ข้อมูลเข้า (Input) ไปถึงข้อมูลออก (Output)	52
ภาพที่ 45 คำสั่งสร้างโมเดล Resnet152 และคำสั่งตั้งค่า block ของ Resnet152	53

รายการภาพประกอบ (ต่อ)

ภาพที่ 46 คำสั่งเพื่อใช้งานโมเดล BSRGANs	53
ภาพที่ 47 ตัวอย่างภาพต้นฉบับ (บน) และภาพที่ผ่านการใช้งาน BSRGANs (ล่าง).....	54
ภาพที่ 48 แสดงจำนวนภาพต้นฉบับ ภาพใบหน้าของชุดข้อมูล FMLD แบ่งตามประเภทของใบหน้า และประเภทของชุดข้อมูล	56
ภาพที่ 49 ตัวอย่างชุดข้อมูล FMLD ประเภทสวมหน้ากากถูกต้อง.....	57
ภาพที่ 50 ตัวอย่างชุดข้อมูล FMLD ประเภทไม่สวมหน้ากาก.....	57
ภาพที่ 51 ตัวอย่างชุดข้อมูล FMLD ประเภทสวมหน้ากากไม่ถูกต้อง.....	58
ภาพที่ 52 ตัวอย่างชุดข้อมูล FMLD แถวบนสุดภาพไม่ชัด แถวกลางภาพเล็ก และแถวล่างสุด ภาพมืด.....	58
ภาพที่ 53 ตัวอย่างชุดข้อมูล AFMDK ประเภทสวมหน้ากากถูกต้อง.....	59
ภาพที่ 54 ตัวอย่างชุดข้อมูล AFMDK ประเภทไม่สวมหน้ากาก	60
ภาพที่ 55 ตัวอย่างชุดข้อมูล AFMDK ประเภทสวมหน้ากากไม่ถูกต้อง.....	60
ภาพที่ 56 แสดงรายละเอียดของภาพใบหน้าในชุดข้อมูล FMLD MTCNN ของกลุ่ม Detected (TP), Detected (FP) และ Not Detected.....	61
ภาพที่ 57 แสดงขั้นตอนการได้มาซึ่งชุดข้อมูล FMLD MTCNN, FMLD 5K, AFMDK.....	62
ภาพที่ 58 แสดงจำนวนภาพใบหน้าของแต่ละส่วนของชุดข้อมูลฝึกที่ถูกแบ่งออกเป็นส่วนของชุดข้อมูล FMLD MTCNN แบ่งตามประเภทของใบหน้า	63
ภาพที่ 59 กราฟแสดงประสิทธิภาพของชุดข้อมูล FMLD 5K (2 class) ของโมเดลข้อมูลเข้าเดี่ยวและ โมเดลข้อมูลเข้าหลายค่าที่มีประสิทธิภาพสูงสุด	69
ภาพที่ 60 กราฟแสดงประสิทธิภาพของชุดข้อมูล FMLD 5K (3 class) ของโมเดลข้อมูลเข้าเดี่ยวและ โมเดลข้อมูลเข้าหลายค่าที่มีประสิทธิภาพสูงสุด	69
ภาพที่ 61 กราฟแสดงประสิทธิภาพเฉลี่ย 3 โมเดลของชุดข้อมูล FMLD MTCNN (2 class) ของ โมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่า.....	73
ภาพที่ 62 กราฟแสดงประสิทธิภาพเฉลี่ย 3 โมเดลของชุดข้อมูล FMLD MTCNN (3 class) ของเดี่ยว โมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่า.....	73
ภาพที่ 63 กราฟแสดงประสิทธิภาพของชุดข้อมูล AFMDK (2 class) ของโมเดลข้อมูลเข้าเดี่ยวและ โมเดลข้อมูลเข้าหลายค่าที่มีประสิทธิภาพสูงสุด	75

รายการภาพประกอบ (ต่อ)

ภาพที่ 64 กราฟแสดงประสิทธิภาพของชุดข้อมูล AFMDK (3 class) ของโมเดลข้อมูลเข้าเดี่ยวและ โมเดลข้อมูลเข้าหลายค่า.....	76
ภาพที่ 65 ขั้นตอนการสร้างชุดข้อมูล ในการทดลองใช้การปรับปรุงภาพ (Image Enhancement) โดยใช้ Super-Resolution (SR)	78
ภาพที่ 66 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากถูกต้องที่มีขนาดเล็กกว่า 50 พิกเซลของ ชุดข้อมูล FMLD 5K.....	79
ภาพที่ 67 ตัวอย่างผลการปรับปรุงภาพประเภทไม่สวมหน้ากากที่มีขนาดเล็กกว่า 50 พิกเซลของ ชุดข้อมูล FMLD 5K.....	79
ภาพที่ 68 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากไม่ถูกต้องที่มีขนาดเล็กกว่า 50 พิกเซล ของชุดข้อมูล FMLD 5K.....	80
ภาพที่ 69 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากถูกต้องที่มีขนาดเล็กกว่า 50 พิกเซลของ ชุดข้อมูล AFMDK	80
ภาพที่ 70 ตัวอย่างผลการปรับปรุงภาพประเภทไม่สวมหน้ากากที่มีขนาดเล็กกว่า 50 พิกเซลของ ชุดข้อมูล AFMDK	81
ภาพที่ 71 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากไม่ถูกต้องที่มีขนาดเล็กกว่า 50 พิกเซล ของชุดข้อมูล AFMDK.....	81
ภาพที่ 72 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากถูกต้องที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซลของชุดข้อมูล FMLD 5K	82
ภาพที่ 73 ตัวอย่างผลการปรับปรุงภาพประเภทไม่สวมหน้ากากที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล ของชุดข้อมูล FMLD 5K.....	82
ภาพที่ 74 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากไม่ถูกต้องที่มีขนาดใหญ่กว่าหรือ เทียบเท่า 50 พิกเซลของชุดข้อมูล FMLD 5K.....	83
ภาพที่ 75 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากถูกต้องที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซลของชุดข้อมูล AFMDK.....	83
ภาพที่ 76 ตัวอย่างผลการปรับปรุงภาพประเภทไม่สวมหน้ากากที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซลของชุดข้อมูล AFMDK.....	84
ภาพที่ 77 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากไม่ถูกต้องที่มีขนาดใหญ่กว่าหรือ เทียบเท่า 50 พิกเซลของชุดข้อมูล AFMDK.....	84

รายการภาพประกอบ (ต่อ)

ภาพที่ 78 กราฟประสิทธิภาพ Non_SR และ SR ของโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล FMLD 5K (3 class) ทดสอบโดยภาพขนาดเล็กกว่า 50 พิกเซล.....	88
ภาพที่ 79 กราฟเปรียบเทียบประสิทธิภาพ Non_SR และ SR ของโมเดลข้อมูลเข้าหลายค่าของ ชุดข้อมูล AFMDK (3 class) ทดสอบโดยภาพขนาดเล็กกว่า 50 พิกเซล.....	89
ภาพที่ 80 กราฟประสิทธิภาพ Non_SR และ SR ของโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล FMLD 5K (3 class) ทดสอบโดยภาพขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล	90
ภาพที่ 81 กราฟประสิทธิภาพ Non_SR และ SR ของโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล AFMDK (3 class) ทดสอบโดยภาพขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล	91

บทที่ 1

บทนำ

1.1. ความสำคัญและที่มาของงานวิจัย

การตรวจจับใบหน้าที่สวมหน้ากาก (Face-mask detection) ใช้ตรวจจับใบหน้าที่สวมหน้ากาก และไม่สวมหน้ากาก มีข้อมูลเข้า (Input) เป็นภาพดิจิทัลหรือวิดีโอที่เป็นภาพต่อเนื่อง การตรวจจับใบหน้าที่สวมหน้ากากคือการตรวจจับวัตถุ (Object Detection) ซึ่งเป็นส่วนหนึ่งในคอมพิวเตอร์วิทัศน์ (Computer Vision) [1] ที่เป็นสาขาหนึ่งของปัญญาประดิษฐ์ ต้องอาศัยการแยกแยะวัตถุ (Object Recognition) [2] ที่จะฝึกให้คอมพิวเตอร์รู้จักการแยกแยะวัตถุออกจากภาพแบบดิจิทัล การแยกแยะวัตถุสามารถแบ่งออกเป็นสามกลุ่มหลัก ๆ ได้แก่ 1) การจำแนกข้อมูลภาพ (Image Classification) ที่จะนำภาพมาจำแนกว่าวัตถุในภาพนั้นคืออะไร 2) การจำกัดตำแหน่งของวัตถุ (Object Localization) จะสร้างกรอบภาพ (Bounding Box: BB) เพื่อบอกว่าวัตถุแต่ละวัตถุอยู่ตำแหน่งไหนของภาพ และ 3) การตรวจจับวัตถุ (Object Detection) [3] การทำงานจะเหมือนการตรวจจับวัตถุผสมกับการจำกัดตำแหน่งของวัตถุ ระบุว่าวัตถุแต่ละวัตถุอยู่ตำแหน่งไหนของภาพ และจัดประเภทเป็นอะไร สำหรับการสกัดคุณลักษณะ (Feature Extraction) ในการตรวจจับวัตถุต้องใช้เทคนิคต่าง ๆ มากมาย เช่น การตรวจจับขอบ การตรวจจับมุม ซึ่งเทคนิคต่าง ๆ นี้ต้องใช้ให้เหมาะสมตามลักษณะงาน เช่น สกัดคุณลักษณะเส้นโค้งเพื่อหาวงกลม โดยแต่ละประเภทของวัตถุมีคุณลักษณะจำนวนมากทำให้ใช้เทคนิคในการสกัดคุณลักษณะจำนวนมาก ต้องจัดการตัวแปรมากขึ้น และต้องเลือกเทคนิคเพื่อสกัดคุณลักษณะให้เหมาะสมกับงาน สำหรับการเรียนรู้เชิงลึก (Deep Learning) สามารถช่วยแก้ปัญหาต่าง ๆ เมื่อชุดข้อมูลป้อนเข้าไปในโมเดลของโครงข่ายประสาท (Neural Networks) โมเดลสามารถหาคุณลักษณะได้ นอกจากนี้การหาวัตถุจากภาพสามารถใช้เทคนิคการเลื่อนช่อง (Sliding Window) โดยใช้ช่องขนาดต่าง ๆ เพื่อหาว่าตำแหน่งไหนของภาพมีวัตถุอยู่ซึ่งต้องใช้ทรัพยากรอย่างมาก ส่งผลให้ใช้เวลานาน [4] [5] การเรียนรู้เชิงลึกสามารถใช้แก้ปัญหาการจำแนกใบหน้าที่สวมหน้ากากโดยใช้แนวคิดถดถอยในการจำแนกการสวมหน้ากาก [6] การใช้แนวคิดพื้นที่ในการจำแนกการสวมหน้ากาก [7] และมีการใช้โมเดลที่พัฒนาโดยใช้การเรียนรู้เชิงลึกเพื่อตรวจจับใบหน้าที่ร่วมกับนำใบหน้าที่ได้เป็นข้อมูลเข้าให้โมเดลที่ใช้เพื่อการจำแนกใบหน้าที่สวมหน้ากาก [8] นอกจากนี้โมเดลที่ใช้การเรียนรู้เชิงลึกรับข้อมูลเข้ามากกว่าหนึ่งข้อมูลและข้อมูลเข้าก็ไม่จำเป็นต้องเป็นชนิดเดียวกัน [9]

ในงานวิจัยขั้นนี้ได้ศึกษาและพัฒนาโมเดลเพื่อการรู้จำใบหน้าสวมหน้ากากโดยใช้การเรียนรู้เชิงลึก มีการใช้ข้อมูลเข้ามากกว่าหนึ่งค่าเพื่อเพิ่มประสิทธิภาพของโมเดลให้มีความถูกต้องมากขึ้นในการจำแนกประเภทใบหน้าสวมหน้ากาก และมีการใช้งานเทคนิค Super Resolution (SR) เพื่อแก้ปัญหาภาพมีความคมชัดต่ำ

1.2. วัตถุประสงค์

เพื่อสร้างขั้นตอนวิธีโมเดลการเรียนรู้เชิงลึกที่ใช้ข้อมูลเข้าหลายค่าสำหรับการตรวจจับใบหน้าสวมหน้ากาก

1.3. ประโยชน์ที่คาดว่าจะได้รับ

ได้โมเดลการเรียนรู้เชิงลึกที่ใช้ข้อมูลเข้าหลายค่าสำหรับการตรวจจับใบหน้าสวมหน้ากากที่มีประสิทธิภาพ

1.4. ขอบเขตงานวิจัย

พัฒนาโมเดลการเรียนรู้เชิงลึกที่ใช้ข้อมูลเข้าหลายค่า โดยใช้ข้อมูลการตรวจจับและจำแนกใบหน้าสวมหน้ากาก ซึ่งเป็นชุดข้อมูลมาตรฐาน Face Mask Label Dataset (FMLD) [10] และ Andrewmvd Face Mask Detection Kaggle (AFMDK) [11] ในการฝึกและทดสอบ ประเมินประสิทธิภาพด้วยค่า Accuracy Precision Recall และ F1

1.5. ขั้นตอนและระยะเวลาการดำเนินการ

1.5.1. ระยะเวลาการดำเนินการ

มิถุนายน 2564 – กันยายน 2566

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

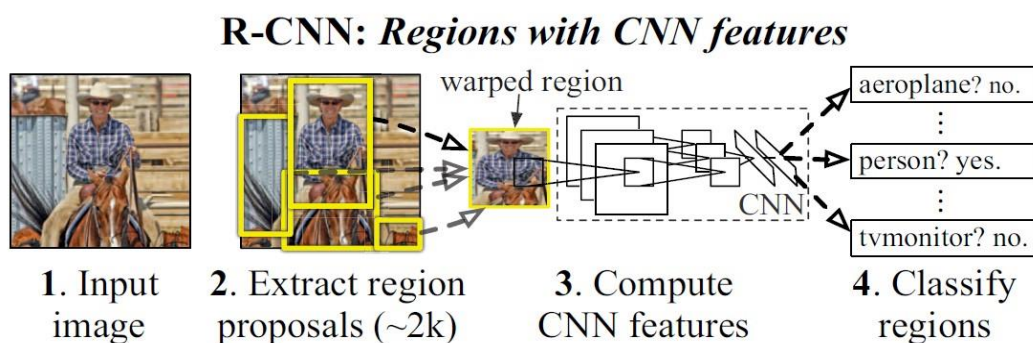
ในบทนี้จะกล่าวถึงทฤษฎีและงานวิจัยต่าง ๆ ที่เกี่ยวข้องกับการพัฒนาการตรวจจับ และการจำแนกใบหน้าสวมหน้ากากโดยใช้ข้อมูลเข้าหลายค่ากับการเรียนรู้เชิงลึก ซึ่งประกอบด้วย การตรวจจับวัตถุ (Object Detection) โครงข่ายประสาทเทียม (Artificial Neural Network) โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks: CNN) โมเดลการเรียนรู้เชิงลึกแบบ ResNet การเตรียมข้อมูลภาพก่อนการประมวลผล (Image Processing) การปรับปรุงภาพ (Image Enhancement) การปรับปรุงภาพโดยใช้โมเดล CodeFormer และ BSRGANs การสกัดคุณลักษณะ (Feature Extraction) การจำแนกประเภท (Classification) การจำกัดตำแหน่ง (Localization) การตรวจจับใบหน้า (Face Detection) โมเดลการเรียนรู้เชิงลึกแบบ Multi-task Cascaded Convolutional Networks (MTCNN) การเรียนรู้แบบถ่ายโอน (Transfer Learning) การวัดประสิทธิภาพของโมเดลการจำแนกประเภทข้อมูล และงานวิจัยที่เกี่ยวข้อง

2.1. การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุเป็นส่วนหนึ่งในคอมพิวเตอร์วิทัศน์ (Computer Vision) ใช้เพื่อหาตำแหน่งของวัตถุ สามารถแบ่งตามแนวคิดที่ใช้ในการแก้ปัญหา ได้แก่ การตรวจจับวัตถุที่ทำงานตามพื้นที่ (Region-based Object Detection) [12] หรือการตรวจจับวัตถุที่ทำงานโดยใช้วิธีการถดถอย (Regression-based Object Detection) [12]

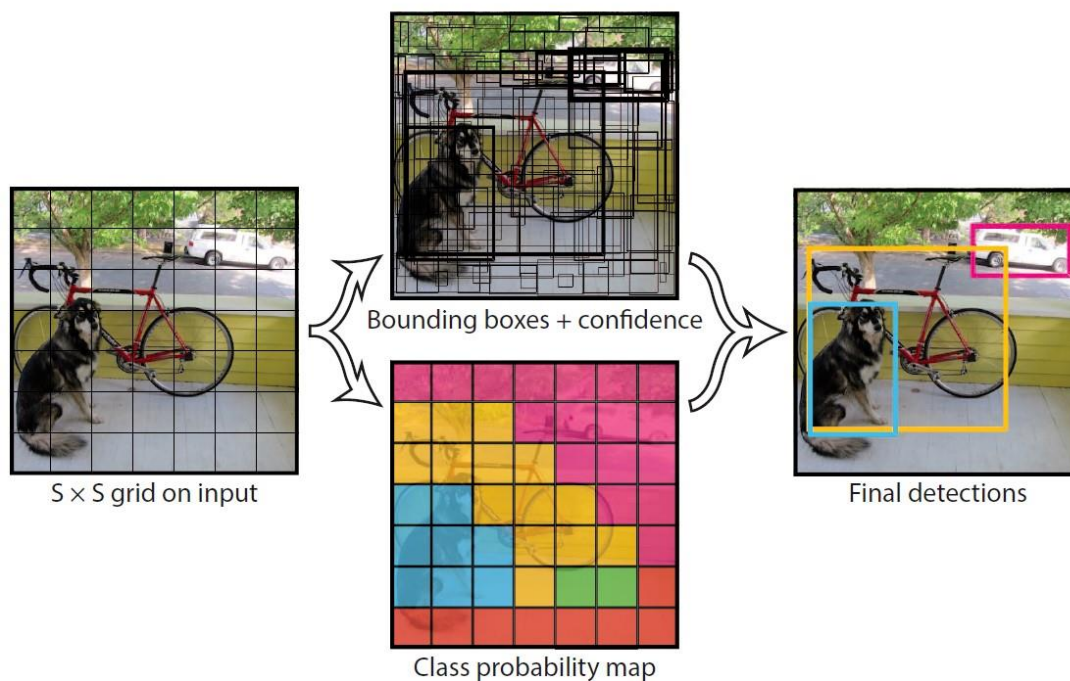
การตรวจจับวัตถุโดยใช้แนวคิดเชิงพื้นที่ จะเป็นการตรวจจับวัตถุในรูปแบบดั้งเดิม คือ สกัดคุณลักษณะเพื่อหาบริเวณที่คาดว่าจะมีวัตถุอยู่ หลังจากนั้นนำกรอบภาพไปสกัดคุณลักษณะ และใช้การถดถอยเพื่อทำให้กรอบภาพจากชั้น Region Proposal พอดีกับขนาดของวัตถุ ตามด้วยจำแนกประเภทของวัตถุ (Classification) ทำให้แนวคิดนี้มีอีกชื่อเรียกหนึ่งคือการตรวจจับวัตถุสองขั้นตอน โมเดลที่ใช้แนวคิดนี้ ได้แก่ โมเดล R-CNN หรืออีกชื่อหนึ่งคือ Region Based CNN การทำงานของโมเดลนี้แสดงดังภาพที่ 1 เริ่มจากขั้นตอนที่เรียกว่า Region Proposal ซึ่งสร้างกรอบภาพในบริเวณที่สนใจ (Region of Interest: RoI) ซึ่งเป็นบริเวณที่คาดว่าจะมีวัตถุอยู่ และตามด้วยนำกรอบภาพไปจำแนกประเภท ในยุคแรก ๆ ของตระกูล R-CNN จะใช้อัลกอริธึม Selective Search เพื่อทำงานในส่วน Region Proposal หลังจากนั้นมีการแทนที่ด้วย Region Proposal Network ซึ่ง

ใช้โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN) เพื่อเสนอบริเวณที่สนใจแทนการใช้ อัลกอริธึม Selective Search [13]



ภาพที่ 1 แสดงขั้นตอนการทำงานของ R-CNN [13]

การตรวจจับวัตถุที่ใช้แนวคิดการถดถอยจะแก้ปัญหการตรวจจับวัตถุโดยทำให้ปัญหาเป็นปัญหาการถดถอย ที่จะรับภาพเป็นข้อมูลเข้า แล้วนำภาพมาแบ่งออกเป็นตาราง จากนั้นสกัดคุณลักษณะเพื่อสร้างกรอบภาพไปพร้อม ๆ กับจำแนกประเภท ซึ่งแตกต่างจากการตรวจจับวัตถุที่ใช้แนวคิดพื้นที่ที่สร้างกรอบภาพที่แสดงถึงบริเวณที่คาดว่าจะมีวัตถุอยู่ หลังจากนั้นจึงนำกรอบภาพนั้นไปสกัดคุณลักษณะเพิ่มเติมและจำแนกประเภท การตรวจจับวัตถุที่ใช้แนวคิดการถดถอยจึงมีอีกชื่อว่าการตรวจจับวัตถุหนึ่งขั้นตอน โมเดลที่ใช้แนวคิดนี้ ได้แก่ โมเดล YOLO และ โมเดล SSD การทำงานของ YOLO จะใช้ Convolutional Neural Network (CNN) 1 ตัว เพื่อสร้างกรอบภาพจำแนก (Predicted Bounding Box) และจำแนกประเภทของกรอบภาพไปพร้อม ๆ กัน การทำงานเริ่มต้นจากการแบ่งภาพออกเป็นตารางดังภาพที่ 2 ช่องไหนมีศูนย์กลางของวัตถุอยู่จะถือว่าช่องนั้นเป็นช่องหลักในการหาวัตถุนั้น โดยแต่ละช่องจะสร้างกรอบภาพออกมาจำนวนหนึ่ง และค่าความเชื่อมั่นของกรอบภาพแต่ละอันเพื่อแสดงถึงโอกาสที่กรอบภาพนั้นจะมีวัตถุอยู่โดยแสดงค่าความเชื่อมั่นต่อประเภทของวัตถุ โดยในแต่ละช่องจะมีค่าความเป็นไปได้ที่บอกว่าวัตถุจะเป็นประเภทไหน [14]



ภาพที่ 2 แสดงการทำงานของ YOLO [14]

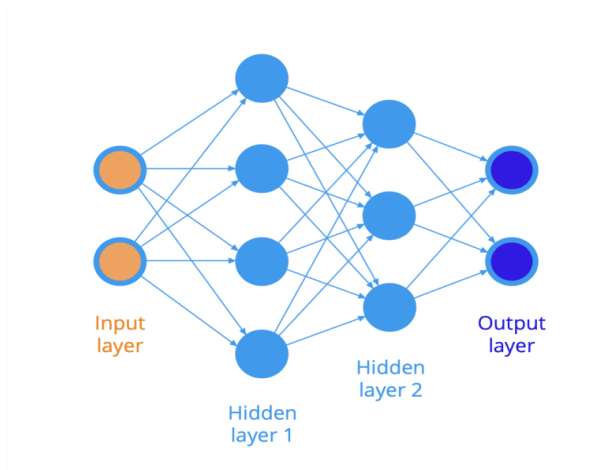
ผลลัพธ์สุดท้ายของการตรวจจับวัตถุในที่นี่จะใช้การตรวจจับวัตถุของ Wang และคณะ [15] ที่ได้นำเสนอการตรวจจับใบหน้าสวมหน้ากากโดยใช้ Pruning YOLO-V4 ร่วมกับ Center Loss ตัวอย่างของผลลัพธ์แสดงดังภาพที่ 3 กรอบภาพที่แสดงถึงบริเวณที่ตรวจจับจะใช้สีเพื่อแสดงถึงประเภทต่าง ๆ เช่น ประเภทสวมหน้ากาก (with_mask) และประเภทไม่สวมหน้ากาก (without_mask) เป็นต้น นอกจากนี้จะมีเลขกำกับแสดงถึงความมั่นใจของโมเดลในการจำแนกในประเภทนั้น ๆ ด้วย



ภาพที่ 3 แสดงผลลัพธ์การทำงานของ [15]

2.2. โครงข่ายประสาทเทียม (Artificial Neural Networks: ANN)

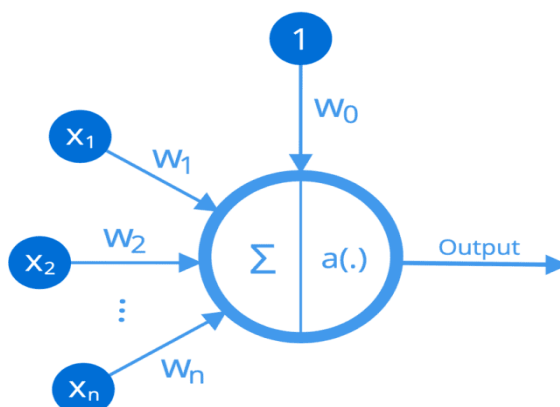
เป็นเทคนิคที่เลียนแบบจากโครงข่ายประสาทของมนุษย์โดยการสร้างโครงข่ายประสาทเพื่อใช้รับข้อมูลเข้า (Input) ข้อมูลจะผ่านกระบวนการต่าง ๆ ในโครงข่ายและออกมาเป็นข้อมูลออก (Output) โดยมีโครงสร้างดังภาพที่ 4



ภาพที่ 4 Neural Networks แบบ Fully-Connected Feedforward [16]

นิวรอน (Neural) หรือโหนด (Node) จะรับข้อมูลเข้าที่อาจเป็นข้อมูลจากชุดข้อมูลหรือข้อมูลจากนิวรอนในชั้นก่อนหน้า และส่งต่อให้นิวรอนในชั้นถัดไป โดยเส้นเชื่อมระหว่างนิวรอน แต่ละตัวจะมีน้ำหนัก w มีค่า n เป็นจำนวนเส้นเชื่อม ซึ่งนิวรอนจะนำข้อมูลเข้า (X) จำนวน n ตัว ไปคูณกับน้ำหนักจากนั้น นำผลลัพธ์ที่จากแต่ละเส้นมารวมกันบวกกับค่าความลำเอียง (Biased: w_0) หลังจากนั้นจะผ่านฟังก์ชัน Activation (a) ออกมาเป็นข้อมูลออกของนิวรอนนั้นดังภาพที่ 5 ซึ่งสามารถคำนวณได้ดังสมการที่ (1)

$$\text{Output} = a \left(\left(\sum_{i=1}^n x_i * w_i \right) + w_0 \right) \quad (1)$$

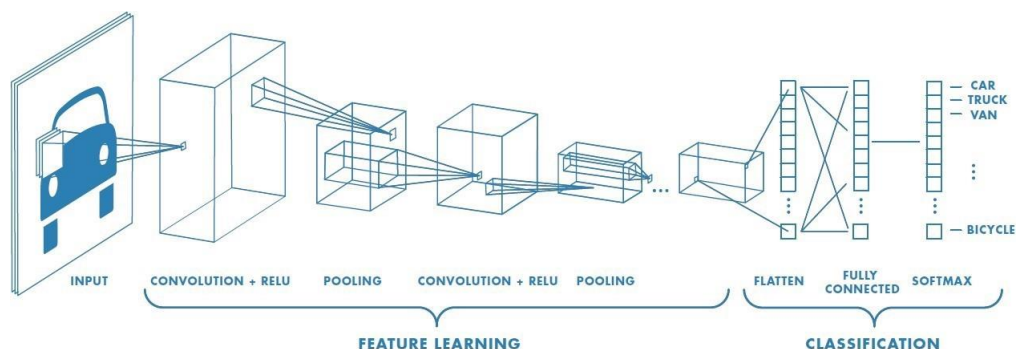


ภาพที่ 5 การทำงานของนิวรอน [16]

เมื่อข้อมูลเข้าถูกป้อนเข้าไปผ่านชั้นต่าง ๆ ในโครงข่ายประสาทเทียมจนถึงชั้นข้อมูลออก จะมีการคำนวณค่าความสูญเสีย (Loss) ที่คำนวณได้โดยเปรียบเทียบข้อมูลออกจาก ANN กับข้อมูลออกจริง ค่าความสูญเสียจะเป็นค่าที่ใช้ในการปรับน้ำหนักให้ฟังก์ชันระหว่างข้อมูลเข้าและข้อมูลออกใกล้เคียงกับที่ต้องการมากที่สุด ANN มีหลายชนิด เช่น Fully-Connected Feedforward ANN เป็น ANN ที่นิวรอนเชื่อมถึงกันหมด โดยนิวรอนทุกตัวในแต่ละชั้นส่งข้อมูลให้นิวรอนในชั้นถัดไปทุกตัว และข้อมูลไหลไปทางเดียว สำหรับ Backpropagation ANN เป็น ANN ที่มีการป้อนค่าย้อนกลับหรืออัตราการเปลี่ยนแปลง (Gradient) ที่คำนวณได้จากค่าความสูญเสีย เพื่อนำไปปรับค่าน้ำหนักเพื่อเพิ่มความถูกต้องให้ข้อมูลออก [13-17]

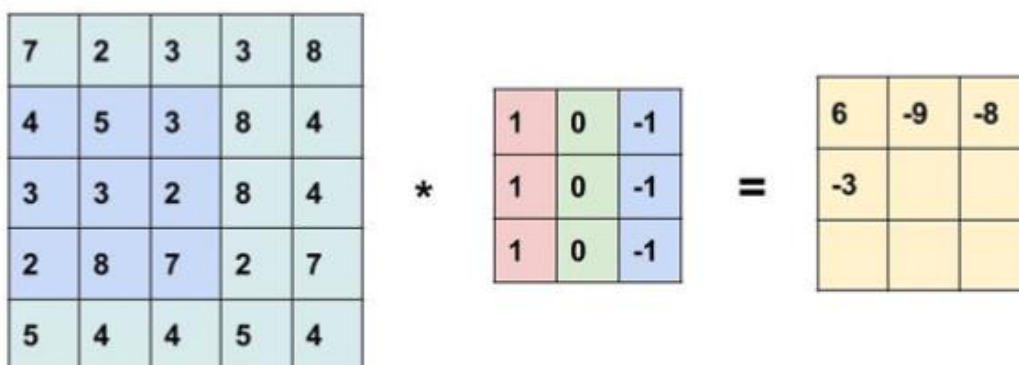
2.3. โครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Networks: CNN)

เทคนิคในการเรียนรู้เชิงลึก CNN เป็น NN ที่เพิ่มชั้น Convolutional เพื่อใช้ในการสกัดคุณลักษณะ (Feature Extraction) ออกมาจากภาพ ทำให้มีความสามารถในการจำแนกประเภทจากภาพ โดย CNN มีโครงสร้างแสดงดังภาพที่ 6 ที่จะมีส่วนข้อมูลเข้า (Input) ตามด้วยส่วนสกัดคุณลักษณะที่ประกอบไปด้วยชั้น Convolutional และชั้น Pooling หลังจากนั้นจะตามด้วยชั้น NN ที่ใช้คุณลักษณะมาจำแนกประเภท



ภาพที่ 6 Convolutional Neural Networks [18]

จุดเด่นของ CNN คือ สามารถเรียนรู้คุณลักษณะได้เอง เนื่องจากแต่ละชั้น Convolutional จะมี Filter หรือ Kernel ที่ค่าน้ำหนักที่สามารถเปลี่ยนแปลงได้ คุณลักษณะเบื้องต้น ได้แก่ เส้นโค้ง เส้นทแยง เส้นแนวตั้ง เส้นแนวนอน เป็นต้น ซึ่งการจะสกัดคุณลักษณะแต่ละตัวจำเป็นต้องใช้ Filter ที่มีน้ำหนักแตกต่างกันไป ตัวอย่างการทำงานของชั้น Convolutional แสดงดังภาพที่ 7 มี Kernel ขนาด 3×3 ที่ทำ Convolution ผ่านภาพขนาด 5×5 พิกเซล ทำให้ได้ผลลัพธ์ออกมาขนาด 3×3



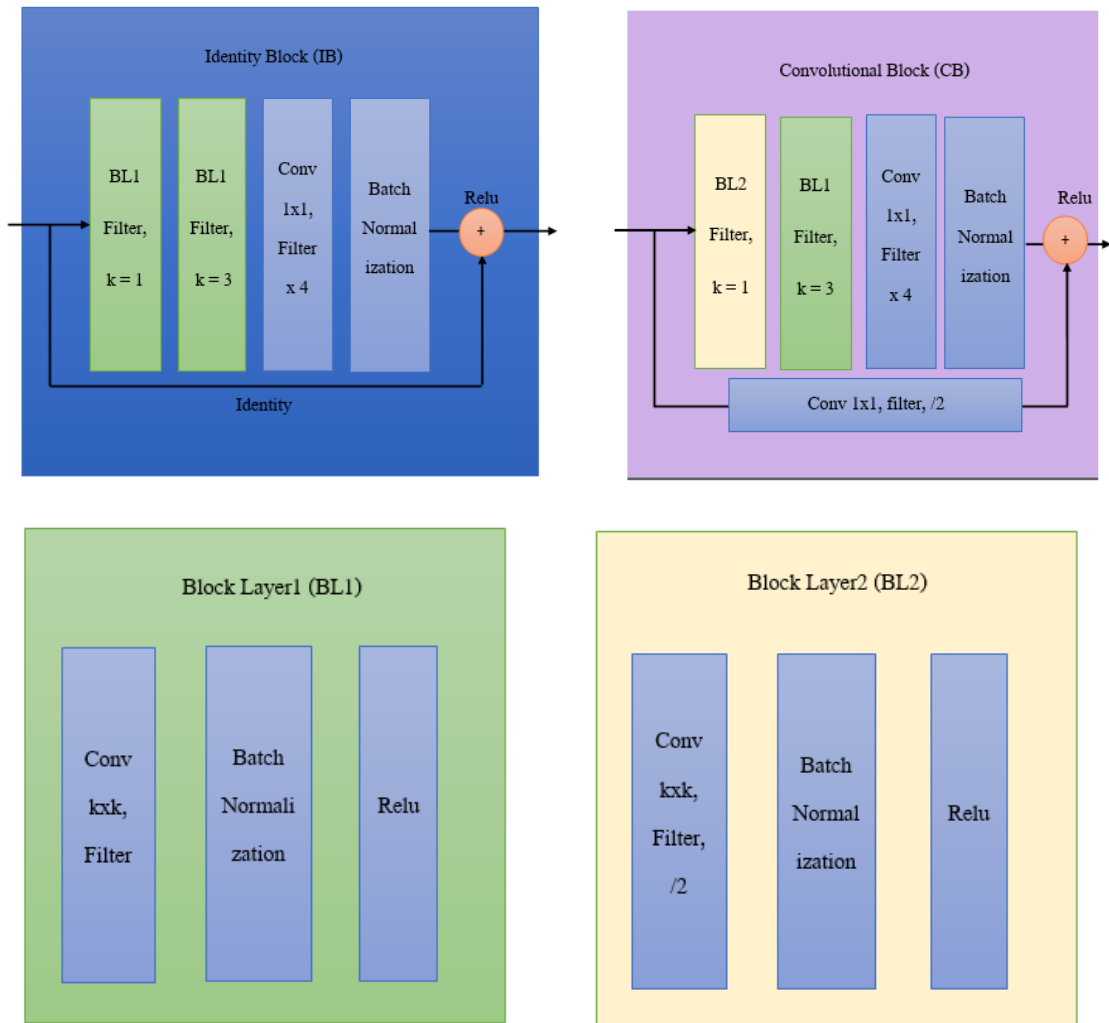
ภาพที่ 7 การทำงานของชั้น Convolutional [18]

การทำงานของ Filter จะดำเนินการคูณเมตริกกับยูนิตแต่ละตัว ตามขนาดของ Filter ทำให้ผลลัพธ์ที่ได้เมื่อผ่าน CNN หนึ่งชั้น มีขนาดลดลงตามการตั้งค่าต่าง ๆ ได้แก่ ขนาดของ Filter ขนาดของ Stride ที่เป็นตัวกำหนดระยะในการเลื่อนของ Filter และค่า Dilate โดย CNN สามารถใช้ Padding เพื่อรักษาขนาดให้ข้อมูลเข้า และข้อมูลออกมีขนาดเท่ากัน เมื่อได้คุณลักษณะแล้วจึงส่งให้ NN เพื่อทำการจำแนกประเภทต่อไป

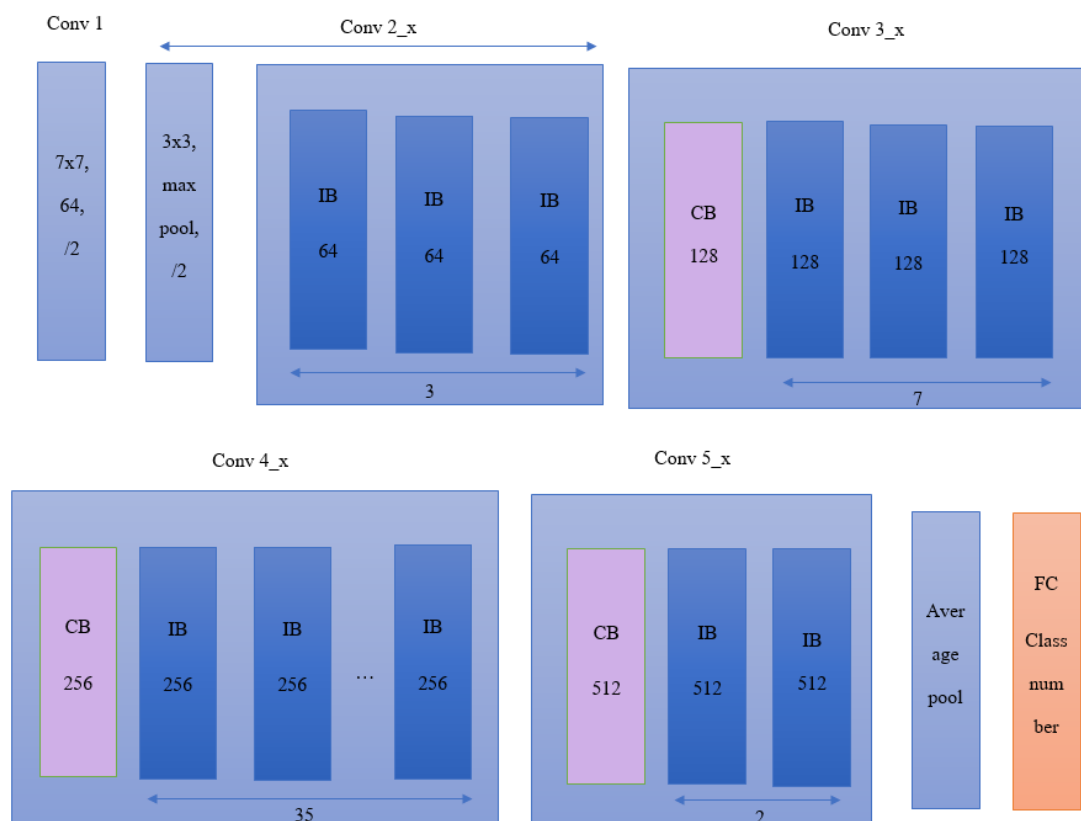
การปรับน้ำหนักของชั้น Convolutional จะใช้ Gradient จากเทคนิค Gradient Descent Gradient คือค่าที่เป็นตัวชี้วัดความสามารถ และบอกทิศทางของโมเดลว่าฟังก์ชัน ที่โมเดลเรียนรู้มีความสามารถใกล้เคียงกับที่ต้องการมากแค่ไหน และต้องปรับน้ำหนักอย่างไร gradient คำนวณได้จากค่าความสูญเสียที่ได้จากการนำค่าที่ทำนายได้กับค่าที่เป็นจริงมาคำนวณ หลักการทำงานของ CNN มีแนวคิดตามพื้นที่การมองเห็นที่ถูกเรียกว่า Reception Field โดยพิกเซลของภาพเป็นยูนิทของ Reception Field การใช้งาน CNN สามารถใช้งานเพื่อลดรูปของภาพให้มีคุณลักษณะลดลง ให้เหลือเฉพาะคุณลักษณะที่สำคัญและง่ายต่อการเข้าใจ [16-19]

2.4. โมเดลการเรียนรู้เชิงลึกแบบ ResNet

Resnet เป็นโมเดลการเรียนรู้เชิงลึกที่มีหมายเลขที่ตามหลังโมเดลที่แสดงถึงจำนวน layer เช่นเดียวกับ VGG โมเดลถูกพัฒนาขึ้นเพื่อแก้ปัญหา Vanishing Gradient ที่เกิดขึ้นใน ANN แบบ Back Propagation ที่จะนำ Gradient จากชั้นสุดท้ายส่งให้ชั้นก่อนหน้า โดยการส่ง Gradient ให้ชั้นก่อนหน้าคุณด้วย Weight Matrix ของชั้นก่อนหน้า ซึ่งเมื่อผ่านชั้นก่อนหน้าไปหลาย ๆ ชั้น จะส่งผลให้ Gradient ลดลง เข้าใกล้ศูนย์เร็วขึ้นส่งผลให้ค่าความแม่นยำถึงจุดอิ่มตัว หรืออาจมีความแม่นยำมีค่าลดลง ResNet จึงแก้ปัญหาโดยใช้ Shortcut Connections ที่สร้าง Identity Map ส่งให้ชั้น ถัดไปหรือสองชั้นถัดไป เพื่อลดความผิดพลาดที่เกิดจากการฝึกฝน ANN ในแต่ละชั้น นอกจากนี้ยังมีการใช้ Global Average Pooling แทน Fully connected (FC) layer ในการจำแนกประเภท เพื่อลดทรัพยากรที่ใช้งาน และแก้ปัญหา Overfitting ของ FC layer [20-21]



ภาพที่ 8 ภาพแสดง Identity Block (IB) Convolutional Block (CB) และ Layer ภายในของ Resnet



ภาพที่ 9 สถาปัตยกรรมของ Resnet152 [21]

Resnet จะมีชั้นจำนวนมาก โดยชั้นแต่ละชั้นจะถูกสร้างเป็นกลุ่มของชั้นแสดงดังภาพที่ 8 กลุ่มของชั้นที่ 1 (Block Layer1: BL1) ประกอบด้วยสามชั้น คือ ชั้น Convolutional ตามด้วยชั้น Batch Normalization และชั้นสุดท้ายคือชั้น Relu Activation สำหรับกลุ่มของชั้นที่ 2 (Block Layer2: BL2) ชั้นแรกของ BL2 จะเป็นชั้น Convolutional ที่มี Stride เป็น 2 แต่ละ BL จะถูกนำไปสร้างเป็น Identity Block (IB) และ Convolutional Block (CB) และจะถูกนำไปสร้างเป็น Resnet ต่อไป ภาพที่ 9 แสดงสถาปัตยกรรม Resnet152 ซึ่งแสดงจำนวนของ IB และ CB ของ Resnet152

2.5. การเตรียมข้อมูลภาพก่อนการประมวลผล (Image Preprocessing)

ขั้นตอนในการเตรียมข้อมูลภาพก่อนให้เหมาะสมสามารถช่วยเพิ่มประสิทธิภาพในการตรวจจับวัตถุได้ ประกอบด้วยการทำให้เป็นมาตรฐาน (Normalization) เช่น การเปลี่ยนค่าในพิกเซลจาก 0 - 255 เป็น 0 - 1 หรือช่วงอื่นตามการใช้งานเพื่อป้องกันปัญหาที่เกิดขึ้นตามมา เช่น เมื่อใช้งาน ANN ถ้าไม่มีการจัดการค่าในพิกเซลของภาพให้เป็นมาตรฐานจะส่งผลให้มีโอกาสเกิดการ

อิมตัวของ Gradient ขึ้นได้ เนื่องจากค่าขนาดใหญ่ของพิกเซลจะถูกนำไปคูณกับค่าน้ำหนักในนิเวรอน ส่งผลให้เมื่อคำนวณ Gradient จะได้ Gradient ขนาดเล็กเกินกว่าจะเกิดการปรับน้ำหนักได้ [22]

การเตรียมข้อมูลภาพก่อนการประมวลผลสามารถแก้ปัญหาต่าง ๆ นอกเหนือจากปัญหาที่กล่าวถึงข้างต้น ได้ แก่ปัญหาการรบกวนของภาพ (Noise) ปัญหาภาพมืดเกินไปหรือภาพมีสีใดสีหนึ่งมากเกินไป และภาพมีความคมชัดต่ำหรือภาพขนาดเล็ก เป็นต้น ซึ่งเทคนิคที่ใช้แก้ปัญหาทั้งหมดนี้เรียกว่าการปรับปรุงภาพ (Image Enhancement)

2.6. การปรับปรุงภาพ (Image Enhancement)

ภาพที่ใช้เป็นข้อมูลเข้าในการจำแนกประเภทที่เป็นภาพจากสถานการณ์จริง คุณภาพของภาพจะไม่สม่ำเสมอ ภาพอาจมีปัญหาต่าง ๆ เช่น ภาพมีความคมชัดต่ำ ภาพขนาดเล็ก ภาพมีการรบกวน ภาพค่อนข้างมืด และอื่น ๆ การปรับปรุงภาพจะเป็นเทคนิคที่นำมาช่วยในการแก้ปัญหาต่าง ๆ ที่ได้กล่าวมา โดยการเพิ่มความคมชัดภาพ ขยายขนาดภาพ ลดการรบกวนในภาพ และปรับสีของภาพ ซึ่งการปรับปรุงภาพอาจใช้เทคนิคที่เกิดจากสูตรคณิตศาสตร์ เช่น การกำจัดการรบกวน (Denoise) เช่น Gaussian Noise ซึ่งสามารถลดการรบกวน แต่ก็ทำให้ภาพมีความคมชัดลดลง และการปรับปรุงภาพแสงต่ำ (low-light Image Enhancement) [23] เป็นต้น สำหรับโมเดลเชิงสถิติ เช่น Histogram Equalization ที่จะเปลี่ยนการกระจายของสีทำให้ภาพที่ได้มีสีที่ตัดกัน (Contrast) มากขึ้น นอกจากนี้ยังมีโมเดลที่แบ่งส่วนประกอบออกเป็นสองส่วน (Decomposition) คือ ความสว่าง (Illumination) และ ความสะท้อนแสง (Reflectance) เพื่อให้สามารถปรับความสว่างได้อย่างอิสระ การเพิ่มความคมชัดของภาพ (Image Sharpening) จะเน้นการเพิ่มความคมชัดของขอบและรายละเอียดของภาพ เช่น Gradient Operator Sharpening หรือ Laplace Operator Sharpening [24] สำหรับการปรับปรุงภาพที่ใช้โมเดลการเรียนรู้เชิงลึก (Deep Learning Model) เช่น โมเดล SRGANs ESRGANs และ BSRGANs เป็นต้น [25]

2.7. การปรับปรุงภาพโดยใช้โมเดล CodeFormer และ BSRGANs

ในส่วนของโมเดลเพื่อใช้ในการปรับปรุงภาพเพื่อเพิ่มความคมชัดเป็นการเตรียมข้อมูลภาพก่อนการประมวลผล เช่น CodeFormer และ BSRGANs

CodeFormer พัฒนาโดย Zhou, Chan, Li & Loy [26] โดยใช้แนวคิดเวกเตอร์ควอนไทซ์เซชัน (Vector Quantization: VQ) กับ Neural Networks แบบ autoencoder โดยการสร้างสร้างคู่มือลงรหัส (Codebook) โดยใช้ภาพคุณภาพสูง (High Quality: HQ) มาป้อนให้กับตัวเข้ารหัส HQ และส่งคุณลักษณะที่ถูกบีบอัดและแทนที่โดยคู่มือลงรหัส หลังจากนั้นนำคุณลักษณะ

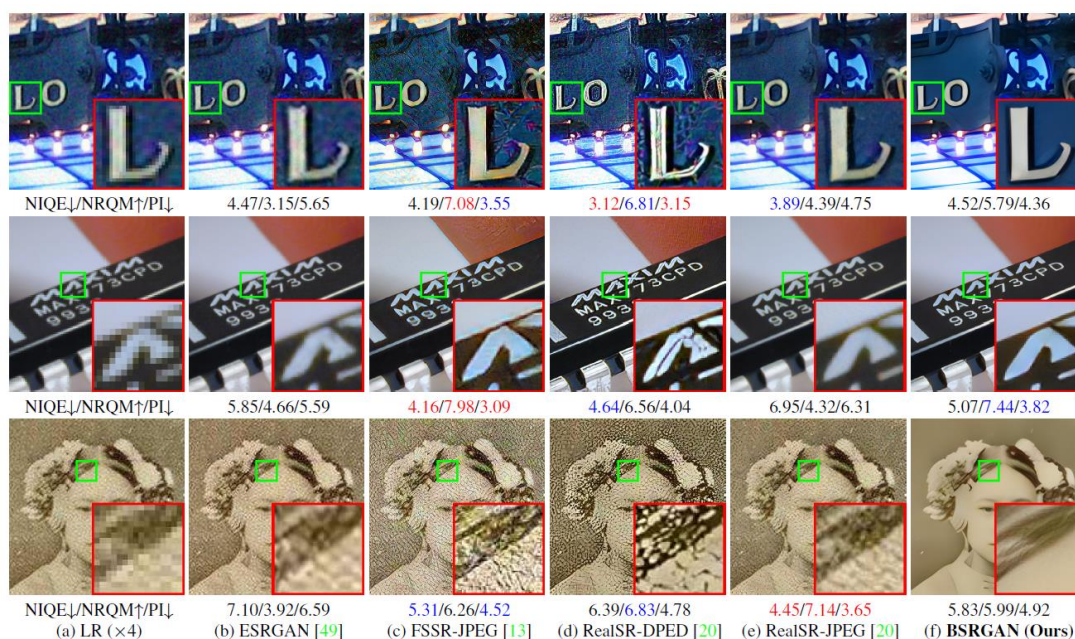
ไปฝึกตัวถอดรหัส HQ และใช้ตัวแปลง (Transformer) เพื่อแก้ปัญหาที่ไม่สามารถใช้คู่มือถอดรหัสได้ เนื่องจากภาพคุณภาพต่ำ (Low Quality: LQ) ไม่สามารถทำนายให้ตรงกับคู่มือถอดรหัส เมื่อลองนำโมเดลนี้มาใช้งานผลลัพธ์ที่ได้ไม่เป็นไปตามที่หวังเนื่องจากเมื่อภาพใบหน้าไม่เห็นจมูกหรือปากโมเดลนี้จะพยายามสร้างจมูกและปากขึ้นมาแสดงดังภาพที่ 10



ภาพที่ 10 ผลลัพธ์จากโมเดล CodeFormer

BSRGANs [27] พัฒนาโดย Zhang, Liang, Gool & Timofte เป็นโมเดลที่ใช้เทคนิค Super-Resolution (SR) และ Generative Adversarial Networks (GANs) โดย SR คือการเพิ่มรายละเอียดของภาพ ที่รวมการแก้ปัญหาภาพเบลอหรือการกำจัดการรบกวนในภาพ GANs คือการพัฒนาโมเดล 2 โมเดล โมเดลแรกรับข้อมูลเข้าเพื่อเรียนรู้รูปแบบของข้อมูลนำไปสร้างข้อมูลใหม่ให้ได้ใกล้เคียงกับกลุ่มของข้อมูลเข้าที่สุ่ม โมเดลที่สองทำนายข้อมูลเข้าว่าเป็นข้อมูลจากชุดข้อมูลจริง หรือข้อมูลที่ได้จากโมเดลแรก การนำ GANs ไปใช้กับเทคนิค SR โมเดลแรกจะทำหน้าที่สร้างภาพให้ใกล้เคียงกับภาพก่อนการลดรายละเอียด และเพิ่มปัญหาต่าง ๆ จนกว่าโมเดลที่สองจะสามารถจำแนกข้อมูลเข้าจากโมเดลแรกให้มีความถูกต้องอยู่ที่ครึ่งหนึ่ง หรือหมายความว่าโมเดลที่สองแยกความแตกต่างระหว่างภาพจากโมเดลแรกและภาพที่เป็นข้อมูลเข้าจริง ๆ ไม่ได้ BSRGANs พัฒนาโดยการนำ Enhanced Super-Resolution Generative Adversarial Networks (ESRGANs) มาเพิ่มความสามารถในการแก้ปัญหาความไม่สมจริงของภาพข้อมูลออกเนื่องจากข้อมูลเข้าที่ใช้ฝึกไม่มีปัญหาที่ครอบคลุมถึงปัญหาที่เกิดขึ้นในภาพจากสถานการณ์จริง ทำให้ได้โมเดลที่ใช้ข้อมูลเข้านี้ในการฝึกยังไม่สามารถตอบสนองต่อปัญหาจริงได้ วิธีแก้ปัญหาคือสร้างโมเดลเฉพาะสำหรับการสร้างข้อมูลเข้าที่ครอบคลุมปัญหาจริงมากขึ้น โดยได้พัฒนาโมเดลการย่อยสลาย (Degradation) เพื่อจำลองผลกระทบ

ของการย่อยสลายทั้ง 3 ได้แก่ เบลอ การดาวน์แซมปลิ่ง (Downsampling) และการรบกวน มีบริเวณมากขึ้นและใกล้เคียงกับสถานการณ์จริงมากขึ้น และใช้การสับเปลี่ยนอย่างสุ่ม ทำให้ภาพรายละเอียดต่ำ (Low Resolution: LR) และใช้ลำดับเทคนิคการย่อยสลายไม่เหมือนกัน ผลลัพธ์การเปรียบเทียบโมเดล BSRGANs กับโมเดลอื่น ๆ แสดงดังภาพที่ 11 และเมื่อทดลองใช้โมเดลได้ผลลัพธ์กับชุดข้อมูลมาตรฐานจะได้ผลลัพธ์แสดงดังภาพที่ 12 [27]



ภาพที่ 11 ผลลัพธ์ของ BSRGANs เมื่อเทียบกับโมเดลอื่น ๆ [27]



ภาพที่ 12 ผลลัพธ์ของโมเดล BSRGANs ด้านบนภาพข้อมูลเข้า และด้านล่างภาพข้อมูลออก

2.8. การสกัดคุณลักษณะ (Feature Extraction)

ขั้นตอนหนึ่งที่ใช้เพื่อสกัดคุณลักษณะออกมาจากชุดข้อมูลเพื่อสร้างเป็นคุณลักษณะใหม่ที่บรรจุลักษณะของข้อมูลที่มีความสำคัญ ช่วยลดความเสี่ยงที่จะเกิด overfitting สามารถเพิ่มความเร็วในการฝึกและเพิ่มความถูกต้อง เทคนิคต่าง ๆ ที่ใช้ในการสกัดคุณลักษณะ เช่น Principle Components Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), Locally Linear Embedding (LLE), t-distributed Stochastic Neighbor Embedding (t-SNE) และ ANN เป็นต้น [28]

2.9. การจำแนกประเภท (Classification)

การจำแนกประเภทของวัตถุในภาพโดยการให้คะแนนความเหมือนของวัตถุในประเภทหนึ่ง ๆ ซึ่งเป็นส่วนหนึ่งของการเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยเทคนิคการจำแนกประเภทถูกแบ่งออกเป็นสองประเภท ได้แก่ Lazy Learner และ Eager Learner [17]

Lazy Learner คือ เทคนิคการจำแนกประเภทที่เก็บข้อมูลทั้งหมดของชุดข้อมูลสำหรับฝึก และนำข้อมูลที่ต้องการจำแนกไปเปรียบเทียบกับข้อมูลที่เก็บไว้ ตัวอย่างเทคนิคที่อยู่ในประเภทนี้ ได้แก่ K-nearest neighbor เป็นต้น

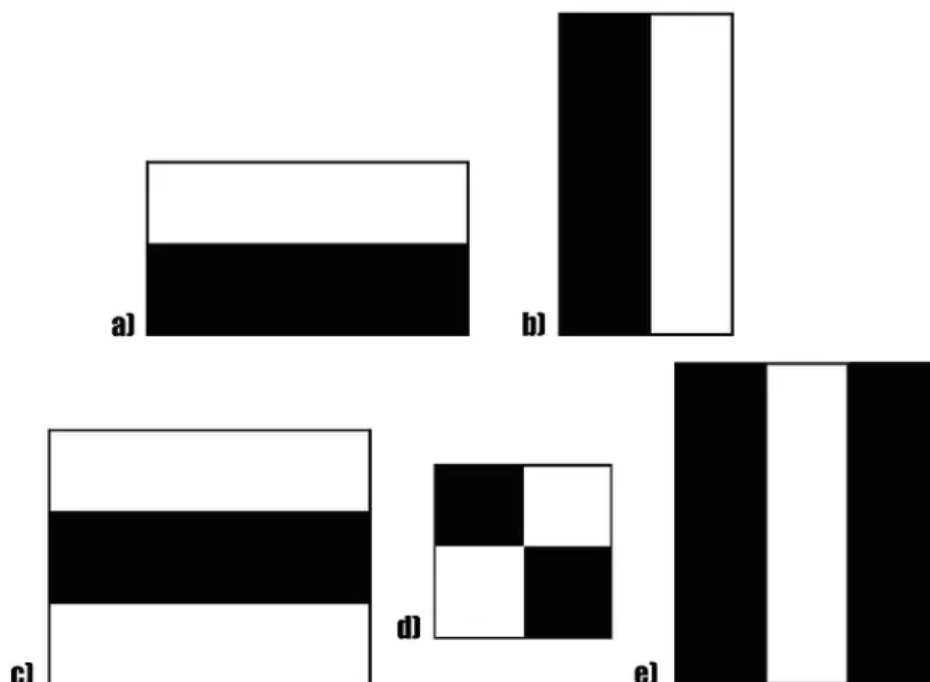
Eager Learner คือ เทคนิคที่นำข้อมูลเข้าไปสร้างเป็นโมเดลที่ใช้ในการจำแนกโดยโมเดลจะสร้างข้อสันนิษฐานในการจำแนกประเภทของข้อมูล ทำให้เทคนิคประเภทนี้ใช้เวลานานแต่ทำงานได้เร็วกว่า Lazy Learner ตัวอย่างเทคนิคที่อยู่ในประเภทนี้ ได้แก่ Support Vector Machine (SVM) หรือ ANN เป็นต้น

2.10. การจำกัดตำแหน่ง (Localization)

การจำกัดตำแหน่งเป็นการหาตำแหน่งของวัตถุในภาพ ใช้กรอบภาพเป็นตัวระบุตำแหน่งของวัตถุ การฝึกใช้กรอบภาพจริง (Ground Truth Bounding Box) ที่แสดงถึงบริเวณที่วัตถุอยู่จริงบนภาพ และข้อมูลออก (Output) เป็นกรอบภาพจำแนก ผลการระบุตำแหน่งจะใช้ค่า Intersection over Union (IoU) เป็นตัววัดว่าบริเวณที่จำแนกได้ มีวัตถุอยู่จริงหรือไม่ กรอบภาพจำแนกใช้การถดถอยขอบเขต (Boundary Regression) เพื่อลดขอบเขตจนถึงบริเวณที่ต้องการ การถดถอยใช้ CNN ในการหาขอบเขตสุดท้าย กรณีการตรวจจับวัตถุที่ใช้แนวคิดเชิงถดถอย จะทำการจำกัดตำแหน่งไปพร้อม ๆ กับการจำแนกประเภท [28-29]

2.11. การตรวจจับใบหน้า (Face Detection)

การตรวจจับใบหน้าใช้เพื่อระบุตำแหน่งของใบหน้าและจมูกกับปาก จึงสามารถใช้เทคนิคการทำใบหน้า เช่น Haar Cascade [30] ที่ใช้ Haar Feature ในการสกัดคุณลักษณะ โดยการหาข้อแตกต่างของของค่าพิกเซลในบริเวณพื้นที่สีขาวและสีดำแสดงดังภาพที่ 13 รวมกับเทคนิค Selective Search และใช้เทคนิคอื่น ๆ เช่น Dlib 68 points Face landmark Detection [31] ในการนำภาพใบหน้าที่หาได้ไปหาภาพบริเวณจมูก และภาพบริเวณปาก โดยหาจุดที่แสดงถึงส่วนสำคัญบนใบหน้าทั้งหมด 68 จุด ที่แสดงถึงโครงหน้า จมูก สันจมูก ปาก ตา และคิ้ว หรือจะใช้เทคนิค หรือโมเดลอื่น ๆ เพื่อการทำใบหน้า จมูกและปากก็ได้

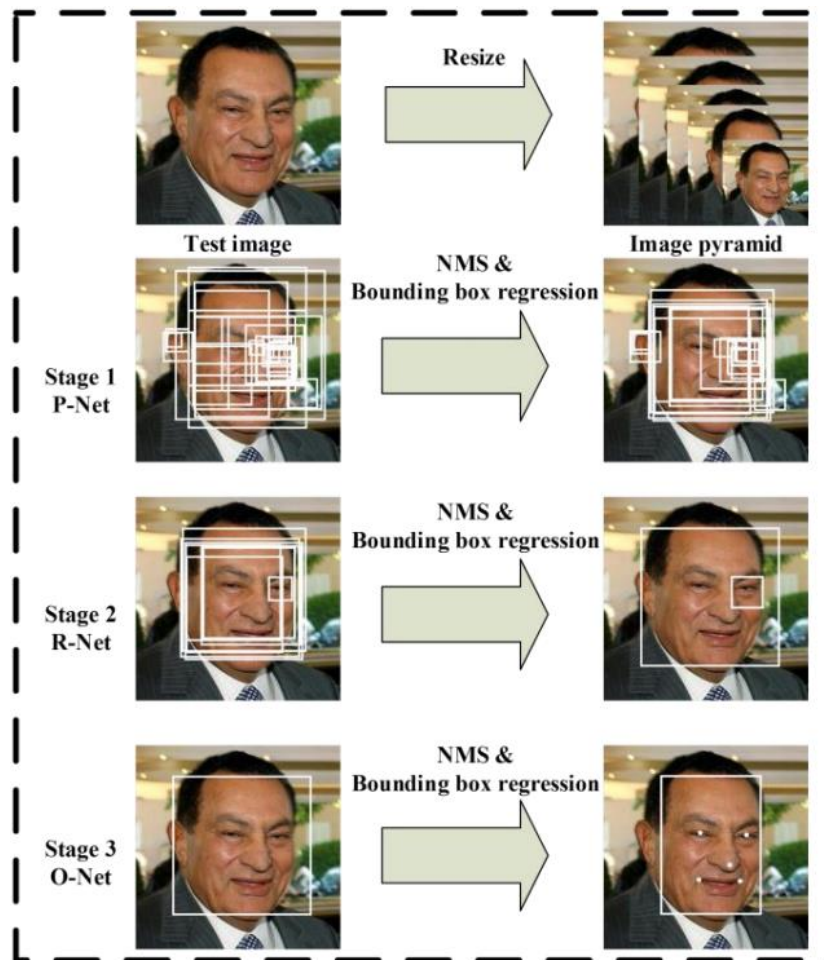


ภาพที่ 13 Haar Feature ที่หาข้อแตกต่างของค่าพิกเซลในพื้นที่สีขาวและสีดำ [32]

Look At Boundary [33] จะใช้ Heatmap รวมกับการถดถอยเพื่อหาจุดสำคัญบนใบหน้า โดยใช้วิธี Hourglass จำนวน 4 ตัวเพื่อสร้าง Heatmap ที่แสดงถึงขอบ และนำ Heatmap ดังกล่าวไปใช้หาจุดสำคัญบนใบหน้า โมเดล Multi-task Cascaded Convolutional Networks (MTCNN) [34] เป็นโมเดลที่มีความสามารถหาใบหน้าพร้อมทั้งตำแหน่งของตา จมูกและปากได้เช่นกัน

2.12. โมเดลการเรียนรู้เชิงลึกแบบ Multi-task Cascaded Convolutional Networks (MTCNN)

MTCNN เป็นโมเดลที่ถูกพัฒนาขึ้นเพื่อใช้ในการตรวจจับใบหน้า โดยมีโมเดล 3 ตัวทำงานร่วมกันแสดงดังภาพที่ 14 ขั้นตอนที่ 1 เริ่มต้นที่ Proposal Network (P-Net) หาบริเวณที่น่าจะมีใบหน้า ขั้นตอนที่ 2 Refinement Network (R-Net) จะนำผลลัพธ์ของ P-Net ไปหาบริเวณที่เป็นไปได้มากที่สุด และสุดท้ายขั้นตอนที่ 3 Output Network (O-Net) จะแสดงกรอบภาพ พร้อมกับตำแหน่งของจุดสำคัญบนใบหน้า เช่น ตา จมูก และปาก เป็นต้น [34]



ภาพที่ 14 ลำดับขั้นตอนการทำงานของ MTCNN [34]

2.13. การเรียนรู้แบบถ่ายโอน (Transfer Learning)

การเรียนรู้แบบถ่ายโอนเป็นเทคนิคหนึ่งที่ใช้ในการเรียนรู้ของเครื่อง (Machine Learning) สามารถนำโมเดลที่เคยฝึกฝน (Pretrained Model) กลับมาใช้งานใหม่ โดยโมเดลที่นำมาใช้จะต้องมีความเกี่ยวข้องกับเป้าหมายที่กำลังทำอยู่ การใช้เทคนิคนี้จะนำน้ำหนักเก่ามาใช้ หรือนำคุณลักษณะจากโมเดลเก่ามาใช้ ทำให้ไม่ต้องเริ่มต้นสร้างโมเดลใหม่ขึ้นมาจากศูนย์ ถ้าโมเดลเก่าเรียนรู้ที่จะสกัดคุณลักษณะเส้นโค้ง ก็จะสามารถนำโมเดลนั้นมาใช้งานกับปัญหาใหม่ที่ต้องตรวจจับสิ่งของที่มีเส้นโค้ง เช่น ลูกบอล ทำให้สามารถฝึกฝนโมเดลใหม่ได้เร็วขึ้นเหมาะสำหรับกรณีที่งานนั้นมีข้อมูลเข้าไม่มากพอ เมื่อใช้เทคนิคนี้จำเป็นจะต้องมีการปรับ (Tuning) น้ำหนักจากโมเดลเก่าให้เหมาะสมกับงานปัจจุบัน ถ้าปรับโมเดลใหม่ได้เหมาะสม รวมทั้งโมเดลที่เคยฝึกฝนมีประสิทธิภาพดีจะทำให้ความแม่นยำสูงขึ้นสามารถฝึกโมเดลใหม่ได้เร็วขึ้นและยังสามารถแก้ปัญหา overfitting ที่เกิดจากการเรียนรู้ข้อมูลเข้าที่ใช้ฝึกฝนมากเกินไป คือทำให้ความแม่นยำสูงในตอนฝึกฝน แต่ตอนใช้งานจริงความแม่นยำต่ำ เนื่องจากข้อมูลเข้าเมื่อใช้งานจริง อาจเป็นข้อมูลที่ไม่เคยพบเจอ ข้อสังเกตในกรณีที่มีข้อมูลมากพอก็ไม่จำเป็นต้องใช้เทคนิคนี้เพราะอาจส่งผลลบจากการถ่ายโอนสิ่งที่ไม่เกี่ยวข้องทำให้น้ำหนักต่าง ๆ เปลี่ยนไปทำให้ค่าความแม่นยำลดลงได้ การเรียนรู้แบบถ่ายโอนแบ่งออกได้เป็นสามแบบ ได้แก่ 1) การเรียนรู้แบบถ่ายโอนโดยอุปนัย (Inductive Transfer Learning) ประเภท 2) การเรียนรู้แบบถ่ายโอนตัวอย่าง (Instance Transfer Learning) และ 3) การเรียนรู้แบบถ่ายโอนตัวแปร (Parameter Transfer Learning) ตัวอย่างโมเดลที่ถูกนำมาใช้งานเพื่อใช้กับการเรียนรู้ถ่ายโอนคือ AlexNet ที่ใช้ในการแข่ง ImageNet ซึ่งมีข้อมูลให้เรียนรู้มากมาย มีประเภทให้จำแนกได้หนึ่งพันประเภท และ MobileNet มีจุดเด่นคือใช้ทรัพยากรน้อย ทำให้สามารถใช้งานในโทรศัพท์หรือระบบฝังตัว (Embedded System) ได้ [35-36]

2.14. การวัดประสิทธิภาพของโมเดลการจำแนกประเภทข้อมูล

ในหัวข้อนี้จะกล่าวถึงเทคนิคการแบ่งชุดข้อมูลและการประเมินประสิทธิภาพของโมเดล ดังรายละเอียดต่อไปนี้

2.14.1. เทคนิคการแบ่งชุดข้อมูลและเทคนิคการประเมินประสิทธิภาพโมเดล

Train/Test Split เป็นเทคนิคพื้นฐานที่ใช้ในการประเมินประสิทธิภาพของโมเดล โดยแบ่งชุดข้อมูลออกเป็น 2 ชุด ได้แก่ ชุดข้อมูลสำหรับฝึก และชุดข้อมูลสำหรับทดสอบตามอัตราส่วนที่ต้องการ ชุดข้อมูลสำหรับฝึกมีไว้เพื่อใช้เป็นข้อมูลเข้าให้แก่โมเดลเพื่อฝึกโมเดลให้ทำงานตามเป้าหมาย และชุดข้อมูลสำหรับทดสอบมีไว้เพื่อทดสอบโมเดลประสิทธิภาพของโมเดลในกรณีที่

พบเจอข้อมูลที่ไม่เคยเจอมาก่อน โดยเทคนิคนี้ไม่เหมาะกับชุดข้อมูลขนาดเล็ก หรือชุดข้อมูลที่มีจำนวนข้อมูลในแต่ละประเภทแตกต่างกันมาก ข้อดีของเทคนิคนี้เมื่อเทียบกับเทคนิคอื่น ๆ คือ ความเร็ว เนื่องจากประหยัดทรัพยากร แต่มีข้อเสียคือถ้าการแบ่งข้อมูลไม่ดีจะส่งผลต่อประสิทธิภาพโมเดลได้ง่าย โดยอัตราส่วนที่นิยมใช้กันในการสอน : การทดสอบคือ 80 : 20, 67 : 33 และ 50 : 50 [37]

2.14.2. การประเมินประสิทธิภาพโมเดล

การประเมินประสิทธิภาพโมเดลสามารถประเมินโดยใช้ค่าที่คำนวณได้จาก Confusion Matrix ซึ่งเป็นตารางแสดงถึงปริมาณของผลลัพธ์การจำแนกของโมเดลที่จำแนกได้ถูกต้องตามประเภทต่าง ๆ (Positive และ Negative) ตัวอย่าง Confusion Matrix แสดงดังตารางที่ 2

ตารางที่ 2 ตาราง Confusion Matrix

		Predicted Class	
		Positive	Negative
Actual Class	Positive	TP	FN
	Negative	FP	TN

เมื่อ TP: True Positive จำแนกประเภทถูกต้องตามที่สนใจ เช่น สวมหน้ากากถูกต้อง

FP: False Positive จำแนกประเภทผิด เช่น จำแนกประเภทเป็นสวมหน้ากากถูกต้อง แต่ประเภทจริงคือสวมหน้ากากไม่ถูกต้อง

FN: False Negative จำแนกประเภทผิด เช่น จำแนกประเภทเป็นสวมหน้ากากไม่ถูกต้อง แต่จริง ๆ แล้ว สวมหน้ากากถูกต้อง

TN: True Negative จำแนกประเภทถูกต้องว่าสวมหน้ากากไม่ถูกต้อง

ตัวชี้วัดประสิทธิภาพสามารถคำนวณได้ดังต่อไปนี้

2.14.2.1. ค่าความถูกต้อง (Accuracy)

ค่าความถูกต้อง คือ ค่าที่แสดงถึงความสามารถในการจำแนกประเภทถูกต้อง ค่าความถูกต้องมีค่าเท่ากับการจำแนกที่ถูกต้องทั้งหมดหารด้วยจำนวนค่าทั้งหมดที่จำแนก โดยสามารถคำนวณโดยใช้สมการที่ (2)

$$\text{Accuracy} = \frac{TP + TN}{P + N} \quad (2)$$

P: Positive จำนวนประเภทที่เป็นเป้าหมายทั้งหมด

N: Negative จำนวนประเภทที่ไม่ใช่เป้าหมายทั้งหมด

2.14.2.2. ค่าความแม่นยำ (Precision)

ค่าความแม่นยำเป็นค่าที่แสดงถึงความแม่นยำในการจำแนกผลลัพธ์อย่างถูกต้องซึ่งคำนวณตามประเภท โดยจะนำผลการจำแนกที่ถูกต้องของประเภทนั้น ๆ (TP) หารด้วยผลการจำแนกที่จำแนกว่าเป็นประเภทนั้น แต่จริง ๆ แล้ว อาจเป็นหรือไม่เป็นประเภทนั้น (TP + FP) ตามสมการที่ (3)

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

2.14.2.3. ค่าความครบถ้วน (Recall)

ค่าความครบถ้วนคือค่าที่แสดงถึงความสามารถในการจำแนกผลลัพธ์อย่างถูกต้อง จากจำนวนข้อมูลในประเภทรูปนั้นๆ ทั้งหมด โดยจะนำผลการจำแนกที่ถูกต้องของประเภทรูปนั้น ๆ (TP) หารด้วยผลการจำแนกที่ทายว่าเป็นประเภทรูปนั้นจริง รวมกับ ผลการจำแนกที่ทายว่าไม่เป็นประเภทรูปนั้น แต่จริง ๆ แล้ว เป็นประเภทรูปนั้น (TP + FN) ตามสมการที่ (4)

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

2.14.2.4. ค่า F1 (F1-score)

ค่า F1 คือค่าที่ใช้วัดประสิทธิภาพของโมเดลในการจำแนกประเภท โดยเน้นไปที่ความสามารถในการจำแนกประเภทโดยรวม แตกต่างจากค่าความถูกต้องตรงที่ค่า F1 จะสูงก็ต่อเมื่อมีค่าความแม่นยำ และค่าความครบถ้วนสูง โดยค่า F1 สามารถคำนวณได้ตามสมการที่ (5) - (6)

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (5)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

2.14.2.5. ค่าเฉลี่ยของความแม่นยำ (Average Precision: AP)

ค่าเฉลี่ยของความแม่นยำเป็นค่าที่บอกถึงประสิทธิภาพของโมเดลในการจำแนกใน แต่ละประเภท โดยใช้ค่าความแม่นยำ ประมาณในช่วงในตำแหน่งหนึ่งของค่าความครบถ้วนจากกราฟ แสดงดังภาพที่ 15 ซึ่งจะแบ่งช่วงของค่าความครบถ้วนออกเป็น 11 ส่วน เริ่มตั้งแต่ 0 เพิ่มขึ้นครั้งละ 0.1 ไปถึง 1.0 และนำค่าความแม่นยำประมาณในช่วงของทุกตำแหน่งนั้นมารวมกัน หารด้วยจำนวน ของส่วน ผลลัพธ์ที่ได้เป็นค่า AP [38]

2.14.2.6. ค่าเฉลี่ยของความแม่นยำทั้งหมดของทุกประเภท (Mean Average Precision: mAP)

ค่าเฉลี่ยของความแม่นยำทั้งหมดของทุกประเภทเป็นค่าหนึ่งที่นิยมใช้แสดงถึงประสิทธิภาพของการตรวจจำวัตถุ สามารถคำนวณได้โดยหาค่าเฉลี่ยของค่า AP ของทุก ๆ ประเภท ตามสมการที่ (7) [38-39]

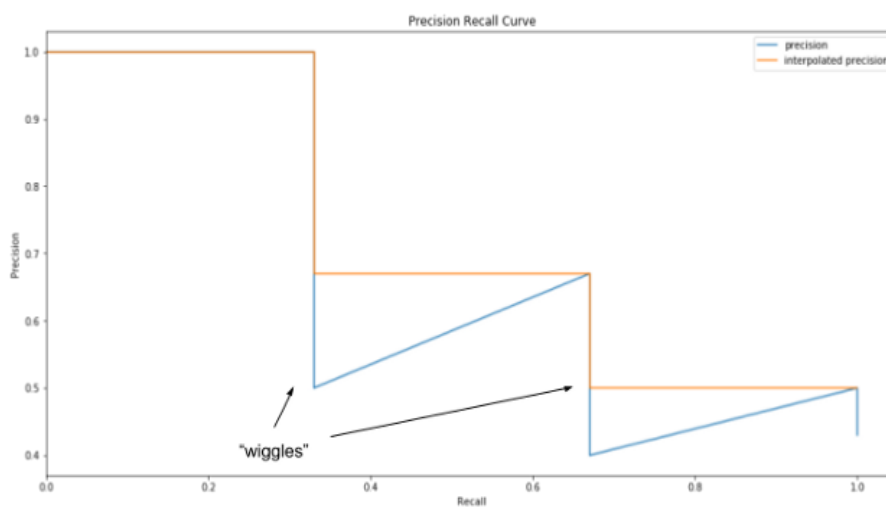
$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (7)$$

2.14.2.7. ค่าความแม่นยำประมาณในช่วง (Interpolated precision)

ค่าความแม่นยำประมาณในช่วงเป็นค่าความแม่นยำสูงสุดในขณะที่ค่าความครบถ้วนเป็นค่าค่าหนึ่ง ใช้เพื่อลดผลกระทบจากความแปรปรวนที่เกิดจากการจำแนกมีปริมาณไม่มากแสดงดังภาพที่ 15 ที่เมื่อนำค่าความแม่นยำไปสร้างเป็นกราฟแสดงดังภาพที่ 16 จะเกิดความแปรปรวนขึ้น จึงใช้ค่าความแม่นยำประมาณในช่วงเพื่อการคำนวณหาค่า AP โดยใช้พื้นที่ใต้กราฟของค่าความแม่นยำประมาณในช่วง และค่าความครบถ้วน โดยจะแบ่งค่าความครบถ้วนเป็น 11 ส่วน เริ่มตั้งแต่ 0 เพิ่มขึ้นครึ่งละ 0.1 จนถึง 1.0 นำค่าความแม่นยำประมาณในช่วงสูงสุดในตำแหน่งค่าความครบถ้วนนั้น มารวมกันแล้วหารด้วยจำนวนส่วนทั้งหมด จากภาพที่ 15 และภาพที่ 16 AP ที่ได้คือ $((4 \times 1) + (3 \times 0.67) + (4 \times 0.5)) = 0.728$

1	TP/FP	Precision	Recall	Precision_inter
2	TP	$1/1 = 1$	$1/3 = 0.33$	1
3	FP	$1/2 = 0.5$	$1/3 = 0.33$	1
4	TP	$2/3 = 0.67$	$2/3 = 0.67$	0.67
5	FP	$2/4 = 0.5$	$2/3 = 0.67$	0.67
6	FP	$2/5 = 0.4$	$2/3 = 0.67$	0.67
7	TP	$3/6 = 0.5$	$3/3 = 1$	0.5
8	FP	$3/7 = 0.43$	$3/3 = 1$	0.5

ภาพที่ 15 ความสัมพันธ์ระหว่างค่าความแม่นยำ ค่าความครบถ้วน และค่าความแม่นยำประมาณ ในช่วง [38]



ภาพที่ 16 กราฟแสดงค่าความแม่นยำ ค่าความครบถ้วน และผลกระทบจากความแปรปรวน [38]

2.15. งานวิจัยที่เกี่ยวข้อง

การตรวจจับการสวมหน้ากากในช่วงที่ผ่านมาได้มีการวิจัยและพัฒนาอย่างแพร่หลาย เนื่องจากสถานการณ์การแพร่ระบาดของไวรัสโควิด 19 ในการศึกษาบทความที่เกี่ยวข้องซึ่งแสดงรายละเอียดต่าง ๆ ประกอบด้วยการตรวจจับบริเวณใบหน้า พบว่ามีการใช้ 2 วิธีหลัก ได้แก่ วิธีที่ 1 การแก้ปัญหาโดยใช้แนวคิดพื้นที่ (Region Based) ที่จะหาบริเวณที่เป็นใบหน้าหรือคาดว่าจะใบหน้าก่อน แล้วจำแนกประเภท โมเดลในกลุ่มนี้จะเป็นโมเดลที่ใช้ Region Proposal ได้แก่ [7] [10] โมเดลในการหาใบหน้า ได้แก่ [8] [40] [41] [42] [43] [44] [45] [46] เป็นต้น วิธีที่ 2 การแก้ปัญหาโดยใช้แนวคิดการถดถอย (Regression Based) ที่จะสกัดคุณลักษณะและจำแนกประเภทไปพร้อม ๆ กับการถดถอยเพื่อจำกัดตำแหน่ง ได้แก่ [15] [47] [48] [49] [50] [6] [51] [52] [53] เป็นต้น นอกจากนี้ในส่วนของการเรียนรู้ยังสามารถใช้งานเทคนิคการเรียนรู้ถ่ายโอน (Transfer Learning) เพื่อลดระยะเวลาในการฝึกในกรณีที่ประสิทธิภาพไม่ลดลงจากการใช้เทคนิคการเรียนรู้ถ่ายโอน ได้แก่ [43] [44] [54] [55] [56] เป็นต้น สำหรับการทำงานของระบบสามารถทำงานได้ในลักษณะของเวลาจริง (Real Time) เมื่อผู้พัฒนาเน้นการใช้งานจริงมากกว่าประสิทธิภาพในการจำแนกประเภท โดยใช้ภาพจากกล้องและมีการวัดความเร็วการทำงาน ได้แก่ [57] [58] รายละเอียดการจำแนกตามประเด็นต่าง ๆ ตั้งแต่ปี 2020 ถึง 2023 แสดงในตารางที่ 3

ตารางที่ 3 แสดงรายละเอียดบทความเกี่ยวกับการตรวจจับใบหน้าสวมหน้ากากที่ได้ศึกษามา

Authors	Year	Region based	Regression based	Use Transfer Learning	Real time
Buciu [40]	2020	✓	✗	✗	✗
Kodali, & Dhanekula [41]	2021	✓	✗	✗	✗
Batagelj, Peer, Štruc, & Dobrišek [42]	2021	✓	✗	✗	✗
Abbasi, Abdi, & Ahmadi [43]	2021	✓	✗	✗	✓
Sakshi, Gupta, Singh Yadav, & Kumar [44]	2021	✓	✗	✗	✓
Gathani, & Shah [7]	2020	✓	✗	✓	✗
Liu, & Ren [57]	2021	✓	✗	✓	✗
Zhang, Han, Chun, & Chen [10]	2021	✓	✗	✓	✗
Podbucki, Suder, Marciniak, & Dabrowski [58]	2020	✓	✗	✓	✓

Authors	Year	Region based	Regression based	Use Transfer Learning	Real time
Srinivasan, Rujula Singh, Biradar, & Revathi [8]	2021	✓	✗	✓	✓
Jeny, Shraddha, Ashritha, Sai, & Naveen [45]	2021	✓	✗	✓	✓
Lin [46]	2022	✓	✗	✗	✗
Shuangyan, & Huayong [47]	2023	✗	✓	✗	✗
Vijitkunsawat, & Chantngarm [48]	2020	✗	✓	✗	✓
Han. Huang, Kuerban, Yan, & Fu [49]	2020	✗	✓	✗	✓
Youssry & Khattab [50]	2022	✗	✓	✗	✓
Venkateswarlu, Kakarla, & Prakash [54]	2020	✗	✓	✓	✗
Oumina, El Makhfi, & Hamdi [55]	2020	✗	✓	✓	✗
Reddy, Nandini, Mamatha, Reddy, & Vishant [56]	2021	✗	✓	✓	✗
Bhuiyan, Khushbu, & Islam [6]	2020	✗	✓	✓	✓
Yang, Feng, Jin, Lei, Gui, & Wang [51]	2020	✗	✓	✓	✓
Sanjaya, & Rakhmawan [52]	2020	✗	✓	✓	✓
Wang, Chen, Wei, & Ling [15]	2020	✗	✓	✓	✓
Karim Sujon, Hossain, Al Amin, Bepery, & Rahman [53]	2022	✗	✓	✓	✓

สำหรับการตรวจจับใบหน้าโดยใช้แนวคิดเชิงพื้นที่ (Region Based) ตั้งแต่ปี 2020 ถึง 2022แสดงรายละเอียดดังตารางที่ 4 เช่นการใช้ Region Proposal หรือโมเดลการเรียนรู้เชิงลึกเพื่อหาใบหน้าจากภาพ [7] [10] สำหรับโมเดลพื้นฐานในการจำแนกประเภทมีหลายชนิด เช่น ResNet50, VGG16, MobileNetV2 และ Custom CNN บางโมเดลใช้การตรวจจับบริเวณที่สนใจ (ROI Detection) และมีการใช้โมเดลที่หลากหลาย เช่น MTCNN, SSD และ YOLOV4 เป็นต้น [53]

ตารางที่ 4 การตรวจจับใบหน้าโดยใช้แนวคิดพื้นที่แสดงโมเดล และการตรวจจับบริเวณที่สนใจ

Authors	Year	Dataset	Model	ROI Detection
Gathani, & Shah [7]	2020	Custom dataset medical masks dataset image for mask, search for no mask	ResNet50	-
Podbucki, Suder, Marciniak, & Dabrowski [58]	2020	Custom dataset	Custom CNN	MTCNN, Haar Cascade
Buciu [40]	2020	Custom dataset	Color information	SSD
Zhang, Han, Chun, & Chen [10]	2021	MAFA + Search	VGG16	-
Srinivasan, Rujula Singh, Biradar, & Revathi [8]	2021	Custom dataset	ResNet50, Xception, MobileNetV2	DSFD, RetinaNetMobileNet
Kodali, & Dhanekula [41]	2021	MAFA + Others	Custom CNN	Haar Cascade
Jeny, Shraddha, Ashritha, Sai, & Naveen [45]	2021	Custom dataset	Custom CNN	MTCNN
Abbasi, Abdi, & Ahmadi [43]	2021	Proposed custom dataset, which is from MAFA, WIDER FACE + custom Irania	Custom CNN	YOLOV4
Sakshi, Gupta, Singh Yadav, & Kumar [44]	2021	Custom dataset from Real-world Masked Face Recognition Dataset, Face Mask Detection dataset, kaggle and search.	NasNetMobile, Densenet121, MobileNetV2	Custom CNN
Batagelj, Peer, Štruc, & Dobrišek [42]	2021	MAFA + Wider Face = FMLD	Resnet-152	RetinaFace

Authors	Year	Dataset	Model	ROI Detection
Karim Sujon, Hossain, Al Amin, Bepery, & Rahman [53]	2022	Mikolaj Witkowski's Medical Mask dataset on Kaggle + PylmageSearch user "Prajna Bhandary"	Fine-tuned MobileNetV2	Open-CV DNN
Lin [46]	2022	Not imply	Color information	Deep learning model

การตรวจจับใบหน้าโดยใช้แนวคิดเชิงถดถอย (Regression Based) ตั้งแต่ปี 2020 ถึง 2023 แสดงรายละเอียดดังตารางที่ 5 ส่วนใหญ่ใช้โมเดล YOLO และ MobileNet ในการพัฒนาโมเดลตรวจจับใบหน้าสวมหน้ากาก และมีการใช้เทคนิคต่าง ๆ เช่น การใช้ Center loss เพื่อช่วยในการพัฒนาโมเดล [15] [51] และ การใช้ Spatial Separable Convolution และ Feature Enhancement Module (FEM) [49] เป็นต้น [47] นอกจากนี้มีการปรับปรุง YOLOV5 โดยใช้ ShuffleNet, CBAM, GhostNet เพื่อลดทรัพยากรที่ใช้ในการตรวจจับใบหน้าสวมหน้ากากเพื่อใช้งานในอุปกรณ์ที่มีความสามารถในการคำนวณไม่สูง

ตารางที่ 5 การตรวจจับใบหน้าโดยใช้แนวคิดเชิงถดถอยแสดงโมเดลต่าง ๆ ที่ใช้งาน

Authors	Year	Dataset	Model
Bhuiyan, Khushbu, & Islam [6]	2020	Custom dataset	YOLOV3
Wang, Chen, Wei, & Ling [15]	2020	medical masks dataset images tfrecords from Kaggle and hikariming virus-maskdataset	Pruning YOLOV4 + Center Loss
Yang, Feng, Jin, Lei, Gui, & Wang [51]	2020	AIZOOTe team's FaceMaskDetection	YOLOV5 + Center Loss
Sanjaya, & Rakhmawan [52]	2020	Kaggle dataset + Real-World Masked Face dataset (RMFD)	MobileNetV2

Authors	Year	Dataset	Model
Vijitkunsawat, & Chantngarm [48]	2020	Prajna Bhandary dataset	SVM, KNN, MobileNet
Han. Huang, Kuerban, Yan, & Fu [49]	2020	Proposed custom dataset COVID-19-Mask	SSD + spatial separable convolution + Feature Enhancement Module (FEM)
Venkateswarlu, Kakarla, & Prakash [54]	2020	1 chandrikadeb7 Face-Mask-Detection, 2 prajnasb observations	MobileNet + Global Pooling
Oumina, El Makhfi, & Hamdi [55]	2020	prajnasb observations face mask dataset	VGG19, Xception, MobileNet + SVM, KNN
Reddy, Nandini, Mamatha, Reddy, & Vishant [56]	2021	Custom dataset	MobileNetV2
Liu, & Ren [57]	2021	Andrewmvd Face Mask Detection Kaggle	Faster R-CNN, YOLOV3, YOLOv3 Weighted Loss, MobileNetV3 for CNAP, YOLONano for CNAP
leamsaard, Charoensook, & Yammen [59]	2021	Andrewmvd Face Mask Detection Kaggle	YOLOV5
Yousry & Khattab [50]	2022	Face Mask Detection (FMD)	YOLOV5
Shuangyan, & Huayong [47]	2023	Custom dataset	Improved YOLOV5 (ShuffleNet on Backbone, CBAM and GhostNet on Neck)

การวัดประสิทธิภาพโมเดลที่ใช้ในการตรวจจับวัตถุจะใช้ค่าความถูกต้อง (Accuracy) ในการวัดประสิทธิภาพการจำแนกประเภทของโมเดลแสดงดังตารางที่ 6

ตารางที่ 6 ผลลัพธ์ของบทความที่ใช้หน่วยวัดประสิทธิภาพเป็นค่าความถูกต้องของชุดข้อมูลต่าง ๆ

Authors	Year	Dataset	Accuracy
Bhuiyan, Khushbu, & Islam [6]	2020	Custom dataset	96.00
Oumina, El Makhfi, & Hamdi [55]	2020	prajnasb observations face mask dataset	97.10
Venkateswarlu, Kakarla, & Prakash [54]	2020	1 chandrikadeb7 Face-Mask-Detection, 2 prajnasb observations	99.48/100.00 (ใช้ชุดข้อมูลสองชุด)
Sanjaya, & Rakhmawan [52]	2020	Kaggle dataset + Real-World Masked Face dataset (RMFD)	92.00
Podbucki, Suder, Marciniak, & Dabrowski [58]	2020	Custom dataset	75.00
Buciu [40]	2020	Custom dataset	97.25
Yang, Feng, Jin, Lei, Gui, & Wang [51]	2020	AIZOOTe team's FaceMaskDetection	97.90
Srinivasan, Rujula Singh, Biradar, & Revathi [8]	2021	Custom dataset	93.20
Jeny, Shraddha, Ashritha, Sai, & Naveen [45]	2021	Custom dataset	97.20
Abbasi, Abdi, & Ahmadi [43]	2021	Proposed custom dataset, which is from MAFA (4,066), WIDER FACE (3,894 total), custom Irania	99.50
Sakshi, Gupta, Singh Yadav, & Kumar [44]	2021	custom dataset from Real-world Masked Face Recognition Dataset, Face Mask Detection dataset, kaggle and search.	99.48
Batagelj, Peer, Štruc, & Dobrišek [42]	2021	MAFA + Wider Face = FMLD	98.93
Kodali, & Dhanekula [41]	2021	MAFA + Others	96.40
Karim Sujon, Hossain, Al Amin, Bepery, & Rahman [53]	2020	Mikolaj Witkowski's Medical Mask dataset on Kaggle + PyImageSearch user "Prajna Bhandary"	98.60
Lin [46]	2022	Not imply	~95.00

การวัดประสิทธิภาพโดยใช้ mAP แสดงถึงประสิทธิภาพของโมเดลในการจำกัดตำแหน่งของวัตถุ และการจำแนกประเภท ตั้งแต่ปี 2020 ถึง 2023 ดังแสดงในตารางที่ 7 มีการกำหนดค่า mAP@0.5 กล่าวคือ IoU มีค่าเท่ากับ 0.5 หมายถึงต้องมีพื้นที่ทับซ้อนกันของกรอบภาพจำแนก และกรอบภาพจริงมากกว่าหรือเท่ากับ 50% การจำแนกประเภทจึงจะถือว่าจำแนกได้ถูกต้อง

ตารางที่ 7 ผลลัพธ์ของบทความที่ใช้หน่วยวัดประสิทธิภาพเป็น mAP ของชุดข้อมูลต่าง ๆ

Authors	Year	Dataset	mAP
Bhuiyan, Khushbu, & Islam [6]	2020	Custom dataset	96.00
Gathani, & Shah [7]	2020	custom dataset medical masks dataset image for mask, search for no mask	85.82
Wang, Chen, Wei, & Ling [15]	2020	medical masks dataset images tfrecords from Kaggle and hikariming virus-maskdataset	80.75
Han. Huang, Kuerban, Yan, & Fu [49]	2020	Proposed custom dataset COVID-19- Mask	90.90
Liu, & Ren [57]	2021	MAFA + Wider Face = FMLD	90.75 +- 0.99
Batagelj, Peer, Štruc, & Dobrišek [42]	2021	MAFA + Wider Face + Others	98.00
Shuangyan, & Huayong [47]	2023	Custom dataset	93.20
Youssry & Khattab [50]	2022	Face Mask Detection (FMD)	84.80

บทที่ 3

วิธีการพัฒนาโมเดลข้อมูลเข้าหลายค่า

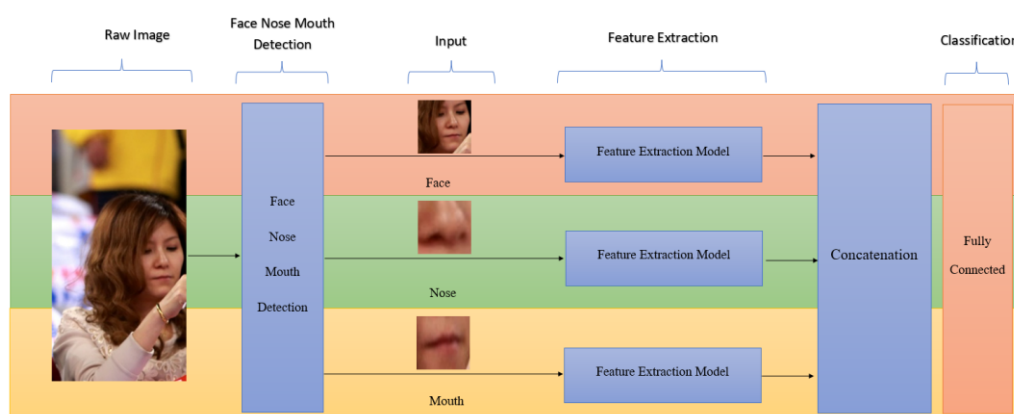
เนื้อหาในบทนี้จะอธิบายวิธีการพัฒนาโมเดลข้อมูลเข้าหลายค่าที่ใช้ในการตรวจจับใบหน้าสวมหน้ากาก การพัฒนาโมเดลข้อมูลเข้าหลายค่าเกิดขึ้นเนื่องจากผู้วิจัยเห็นถึงข้อแตกต่างระหว่างประเภททั้ง 3 ประเภทของใบหน้าสวมหน้ากาก ได้แก่ ประเภทแรก คือ สวมหน้ากากถูกต้อง (with_mask) จะมีการสวมหน้ากากปิดทั้งบริเวณจมูก และปาก ไม่สามารถเห็นทั้งสองส่วนนี้ได้ ประเภทที่สองไม่สวมหน้ากาก (without_mask) เมื่อไม่มีอะไรมาบังใบหน้าจะเห็นบริเวณจมูกและปากชัดเจน และประเภทที่สามสวมหน้ากากไม่ถูกต้อง (wear_mask_incorrect) ที่เป็นไปได้ตั้งแต่เห็นทั้งบริเวณจมูกและปาก เห็นอย่างใดอย่างหนึ่ง หรือเห็นแค่บางส่วน ผู้วิจัยจึงพัฒนาโมเดลข้อมูลเข้าหลายค่าเพื่อรับภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปากเพื่อใช้ในการจำแนกประเภทการสวมหน้ากาก ได้ทดลองออกแบบโมเดลที่สามารถรับข้อมูลเข้าหลายค่าเป็นภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก ออกแบบสถาปัตยกรรมที่ทำหน้าที่ตรวจจับภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก และโมเดลที่สกัดคุณลักษณะและจำแนกประเภท

ในบทนี้จะกล่าวถึง การออกแบบสถาปัตยกรรมและขั้นตอนวิธีสร้างและทดสอบโมเดล การตรวจจับใบหน้า จมูก และปากโดยใช้ MTCNN การสกัดคุณลักษณะ (Feature Extraction) และการจำแนกประเภท (Classification) และการปรับปรุงภาพโดยโมเดลการเรียนรู้เชิงลึกแบบ BSRGANs

3.1. การออกแบบสถาปัตยกรรมและขั้นตอนวิธีสร้างและทดสอบโมเดล

การตรวจจับใบหน้าที่กล่าวมาจะใช้ข้อมูลจากภาพใบหน้าเพียงอย่างเดียว หรือสกัดข้อมูลจากภาพใบหน้า กรณีการจำแนกประเภทของใบหน้าสวมหน้ากากสังเกตได้ว่าประเภทสวมหน้ากากถูกต้องจะมีการสวมหน้ากากเรียบร้อยปิดทั้งบริเวณจมูกและปาก ในขณะที่ประเภทไม่สวมหน้ากากจะเห็นบริเวณจมูกและปากชัดเจน เมื่อไม่มีอะไรมาบังใบหน้า และประเภทสวมหน้ากากไม่ถูกต้องที่อาจเหมือนประเภทไม่สวมหน้ากากที่เห็นบริเวณจมูกและปาก แต่บางส่วนของใบหน้าถูกปิดหรือเห็นแค่บริเวณจมูกปากอย่างใดอย่างหนึ่ง ผู้วิจัยจึงสร้างโมเดลข้อมูลเข้าหลายค่าเพื่อรับภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปากเพื่อใช้ในการจำแนกประเภทการสวมหน้ากาก

สถาปัตยกรรมของโมเดลข้อมูลเข้าหลายค่าเริ่มต้นจากการนำภาพต้นฉบับส่งไปให้โมเดลที่ใช้ตรวจจับใบหน้าแสดงดังภาพที่ 17 ภาพต้นฉบับ (Raw Image) จากชุดข้อมูล จะถูกส่งไปให้โมเดลที่ใช้ตรวจจับใบหน้าที่มีความสามารถตรวจจับภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก (Face Nose Mouth Detection) หลังจากนั้นภาพทั้งหมดที่ได้จะถูกส่งไปสกัดคุณลักษณะ (Feature Extraction) และคุณลักษณะจากแต่ละภาพจะถูกรวมเข้าด้วยกันและส่งไปใช้ในการจำแนกประเภท (Classification) ในชั้น Fully Connected (FC)



ภาพที่ 17 สถาปัตยกรรมที่ออกแบบโมเดลการตรวจจับใบหน้าสวมหน้ากากข้อมูลเข้าหลายค่า

รายละเอียดการทำงานเป็นไปดัง Algorithm 1 สำหรับชุดข้อมูลฝึก ขั้นตอนที่ 1 จะอ่านข้อมูลภาพ (Raw Image) และนำข้อมูลที่อ่านได้ไปใช้ในขั้นตอนที่ 2 เพื่อตรวจจับใบหน้าโดยใช้โมเดล Multi-task Cascaded Convolutional Networks (MTCNN) ทำให้ได้ภาพข้อมูลเข้าเป็นภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก ขั้นตอนที่ 3 – 5 ปรับขนาดภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปากให้มีขนาด 224×224 พิกเซล เพื่อให้ตรงตามขนาดข้อมูลเข้าของโมเดล Resnet 152 ขั้นตอนที่ 6 ใช้คำสั่ง (แสดงดังภาพที่ 18) เพื่อสร้างตัวโหลดชุดข้อมูลที่ปรับค่าของพิกเซลให้อยู่ในช่วง 0-1 ขั้นตอนที่ 7 ใช้คำสั่ง (แสดงดังภาพที่ 19 และภาพที่ 20) เพื่อโหลดชุดข้อมูลสำหรับฝึกและชุดข้อมูลสำหรับทดสอบ ชุดข้อมูลสำหรับฝึกจะใช้คำสั่งให้สุ่มข้อมูลเข้า และตั้งค่า Seed สำหรับก่อนการสุ่มเพื่อให้ภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก สุ่มได้ภาพที่มาจากภาพต้นฉบับเดียวกัน หลังจากนั้นสร้างตัวป้อนข้อมูลแบบหลายค่าให้กับโมเดล (แสดงดังภาพที่ 21) และใช้คำสั่ง (แสดงดังภาพที่ 22) เพื่อสร้างตัวแปรเก็บข้อมูลเข้าสำหรับส่งให้โมเดลข้อมูลเข้าหลายค่า ขั้นตอนที่ 8 สร้างโมเดล Resnet 152 สำหรับการสกัดคุณลักษณะจากชุดข้อมูล และรวมคุณลักษณะที่สกัดได้ในขั้นตอนที่ 9 และนำไปเชื่อมกับชั้น Fully Connected ในขั้นตอนที่ 10 และประกอบเป็นโมเดลข้อมูลเข้าหลายค่าในขั้นตอนที่ 11 และกำหนดรูปแบบการฝึกโมเดลในขั้นตอนที่ 12 โดยใช้ Optimizer เป็น Adam และค่าความสูญเสียที่ใช้ Crossentropy สุดท้ายจะฝึกโมเดลใน

ขั้นตอนที่ 13 จำนวน 200 รอบ ในส่วนของการทดสอบโมเดลในขั้นตอนที่ 14 - 20 เป็นเช่นเดียวกับขั้นตอนที่ 1 -7 ของชุดข้อมูลฝึกแต่เปลี่ยนชุดข้อมูลเป็นชุดข้อมูลสำหรับทดสอบ จากนั้นในขั้นตอนที่ 21 ใช้ชุดข้อมูลทดสอบจำแนกประเภทใบหน้า

ขั้นตอนวิธีที่ใช้ในการสร้างและทดสอบโมเดลข้อมูลเข้าหลายค่าแสดงดัง Algorithm 1

Algorithm 1: Face-Mask Detection and Classification using Multiple Inputs	
Training Process	
1. อ่านข้อมูลภาพ	
2. ตรวจสอบบริเวณภาพใบหน้า โดยใช้ MTCNN	//(หัวข้อ 3.2)
3. ปรับขนาดภาพใบหน้าเป็น 224x224	
4. ตรวจสอบบริเวณจมูกได้ภาพขนาด 50x50 และปรับขนาดเป็น 224x224	
5. ตรวจสอบบริเวณปากได้ภาพขนาด 100x50 และปรับขนาดเป็น 224x224	
6. ปรับช่วงค่าสีให้อยู่ระหว่าง 0 ถึง 1	
7. สำหรับชุดข้อมูลสร้างข้อมูลเข้าที่ประกอบด้วย	//(ภาพที่ 21,28)
กรณี 1 ภาพ คือ ภาพใบหน้า หรือ ภาพบริเวณจมูก หรือ ภาพบริเวณปาก	
กรณี 2 ภาพ คือ ภาพใบหน้าและภาพบริเวณจมูก หรือ ภาพใบหน้าและภาพบริเวณปาก	
หรือ ภาพบริเวณจมูกและภาพบริเวณปาก	
กรณี 3 ภาพ คือ ภาพใบหน้า ภาพบริเวณจมูก และ ภาพบริเวณปาก	
8. สร้างโครงสร้างโมเดล ResNet152 สำหรับใบหน้า จมูก และปาก โดยกำหนดชั้นส่งออก	
คุณลักษณะเป็นชั้น Global Average Pooling	
//(ภาพ 18) (หัวข้อ 3.3)	
9. สร้างชั้นรวมคุณลักษณะใบหน้า จมูก และปากซึ่งเป็นค่าส่งออกจากชั้น Global Average Pooling	//(ภาพ 24)
(ในตอนนี้โครงสร้างโมเดลสำหรับใบหน้า จมูก และปาก ในข้อ 8 จะถูกรวมโครงสร้างเข้าด้วยกัน)	
10. สร้างชั้นแบบ Fully Connected layer ที่มีจำนวนโหนดเท่ากับจำนวนคลาส	//(ภาพ 25)
11. สร้างชั้นนำเข้าแบบข้อมูลเข้าหลายค่าประกอบด้วยภาพใบหน้า จมูก และปาก โดยเชื่อมชั้นนำเข้าของแต่ละโมเดลในข้อ 8	//(ภาพ 26)
12. กำหนดรูปแบบการฝึกโมเดล ได้แก่ ค่า optimizer ค่า loss และการเพิ่มความสามารถในขณะฝึก	//(ภาพ 27,28)

Algorithm 1: Face-Mask Detection and Classification using Multiple Inputs	
13. ฝึกโมเดลโดยใช้ชุดข้อมูลฝึก จำนวน 200 รอบ	//(ภาพ 29)
Testing Process	
14. อ่านข้อมูลภาพ	
15. ตรวจสอบบริเวณภาพใบหน้า โดยใช้ MTCNN	//(หัวข้อ 3.2)
16. ปรับขนาดภาพใบหน้าเป็น 224x224	
17. ตรวจสอบบริเวณจมูกได้ภาพขนาด 50x50 และปรับขนาดเป็น 224x224	
18. ตรวจสอบบริเวณปากได้ภาพขนาด 100x50 และปรับขนาดเป็น 224x224	
19. ปรับช่วงค่าสีให้อยู่ระหว่าง 0 ถึง 1	
20. สำหรับชุดข้อมูลสร้างข้อมูลเข้าที่ประกอบด้วย กรณี 1 ภาพ คือ ภาพใบหน้า หรือ ภาพบริเวณจมูก หรือ ภาพบริเวณปาก กรณี 2 ภาพ คือ ภาพใบหน้าและภาพบริเวณจมูก หรือ ภาพใบหน้าและภาพบริเวณปาก หรือ ภาพบริเวณจมูกและภาพบริเวณปาก กรณี 3 ภาพ คือ ภาพใบหน้า ภาพบริเวณจมูก และ ภาพบริเวณปาก	
21. ป้อนข้อมูลเข้าโมเดลเพื่อจำแนกการใส่หน้ากาก	//(ภาพ 30)

```
idg = ImageDataGenerator(rescale=1. / 255)
```

ภาพที่ 18 คำสั่งเพื่อสร้างตัวโหลดชุดข้อมูลที่ปรับค่าของพิกเซลให้อยู่ในช่วง 0-1

```
trdata = tr_idg.flow_from_directory(directory=dataset_path, batch_size=batch_size, shuffle=True, target_size=(224,224), classes = class_names)
```

ภาพที่ 19 คำสั่งโหลดชุดข้อมูลสำหรับฝึก

```
p_data = p_idg.flow_from_directory(directory=p_path, shuffle=False, target_size=(224,224), classes = class_names)
```

ภาพที่ 20 คำสั่งโหลดข้อมูลสำหรับทดสอบ

```

class CombinedGen():
    def __init__(self, *gens):
        self.gens = gens
        #self.

    def generate(self):
        while True:
            result = []
            i = 0
            y = 0
            for g in self.gens:
                g_v = g.next()
                if i == 0:
                    y = g_v[1]
                    result.append(g_v[0])
                else:
                    result.append(g_v)
                    i = i + 1
            yield result,y

    def __len__(self):
        return len(self.gens[0])

    def __getitem__(self, index):
        x = []
        y = 0
        i = 0
        for g in self.gens:
            data = g.__getitem__(index)
            if i == 0:
                y = data[1]
                x.append(data[0])
            else:
                x.append(data)
                i = i + 1
        return x, y

```

ภาพที่ 21 คำสั่งสร้างตัวป้อนข้อมูลเข้าให้กับโมเดล โดยสามารถป้อนข้อมูลเข้าของภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก


```

cg_tr = 0
if face_module and nose_module and mouth_module:
    cg_tr = CombinedGen(trdata, trdata2, trdata3)
elif face_module:
    if nose_module:
        cg_tr = CombinedGen(trdata, trdata2)
    elif mouth_module:
        cg_tr = CombinedGen(trdata, trdata3)
    else:
        cg_tr = CombinedGen(trdata)
else:
    cg_tr = CombinedGen(trdata2, trdata3)

tr_inputs = cg_tr.generate()

```

ภาพที่ 22 สร้างตัวแปรเก็บข้อมูลเข้าสำหรับป้อนให้โมเดล

หลังจากนั้นจะใช้คำสั่งแสดงดังภาพที่ 23 เพื่อนำข้อมูลออกจากชั้น Global Average Pooling มาเป็นข้อมูลออกของโมเดลย่อยที่อยู่ภายในทั้ง 3 โมเดล พร้อมทั้งตั้งค่าโมเดลให้สามารถฝึกได้ และนำข้อมูลออกของโมเดลย่อยส่งให้ชั้น Concatenate แสดงดังภาพที่ 24 ซึ่งจะส่งข้อมูลออกไปให้ชั้น FC ทำการจำแนกประเภทแสดงดังภาพที่ 25 s_model, s_model2, s_model3 คือ โมเดล Resnet ที่รับข้อมูลเข้าเป็นภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก ตามลำดับ โดยใช้ตัวแปร training เพื่อตั้งค่าชั้นต่าง ๆ ในโมเดลในที่นี่จะตั้งค่า training เป็นจริงเพื่อฝึกโมเดล output_layer_index จะเป็น index เพื่อรับค่าข้อมูลออกจากชั้นตามตัวแปรนี้ รับค่าข้อมูลออกจากชั้น Global Average Pooling ของโมเดล Resnet

```

if face_module:
    s_model_fe1 = tf.keras.Model(inputs=s_model.inputs, outputs=s_model.layers[output_layer_index].output)
    s_model_fe1.trainable = training

if nose_module:
    s_model_fe2 = tf.keras.Model(inputs=s_model2.inputs, outputs=s_model2.layers[output_layer_index].output)
    s_model_fe2.trainable = training

if mouth_module:
    s_model_fe3 = tf.keras.Model(inputs=s_model3.inputs, outputs=s_model3.layers[output_layer_index].output)
    s_model_fe3.trainable = training

```

ภาพที่ 23 คำสั่งสำหรับสร้างเพื่อเตรียมข้อมูลออกของโมเดลข้อมูลเข้าหลายค่า

```

output_list = []
if face_module:
    output_list.append(s_model_fe1.output)

if nose_module:
    output_list.append(s_model_fe2.output)

if mouth_module:
    output_list.append(s_model_fe3.output)

combined = tf.keras.layers.Concatenate()(output_list)

```

ภาพที่ 24 สร้างชั้น Concatenate เพื่อรวมข้อมูลออกของโมเดลที่รับข้อมูลเข้าเป็นภาพใบหน้า ภาพบริเวณจมูกและภาพบริเวณปาก

```

z = Dense(units=class_num, activation=activation_function)(combined)

```

ภาพที่ 25 เชื่อมต่อชั้น Concatenate เข้ากับชั้น FC

คำสั่งแสดงดังภาพที่ 26 จะนำผลลัพธ์ที่ได้จากภาพที่ 25 และข้อมูลเข้าจากโมเดลย่อยมาสร้างเป็นโมเดลข้อมูลเข้าหลายค่าที่สามารถรับข้อมูลเข้าเป็นภาพใบหน้า ภาพบริเวณจมูกและภาพบริเวณปาก หลังจากนั้นจะใช้คำสั่งแสดงดังภาพที่ 27 เพื่อตั้งค่าตัว Optimizer ในที่นี่ใช้ Adam ฟังก์ชันการสูญเสีย เป็น Crossentropy และตั้งค่าหน่วยวัดประสิทธิภาพเป็นค่าความถูกต้อง พร้อมทั้งเพิ่มความสามารถให้ขณะฝึกมีการปรับค่าการเรียนรู้ให้ลดลง เมื่อค่าความสูญเสียคงที่ เพื่อให้โมเดลมีประสิทธิภาพสูงขึ้น เมื่อเทียบกับโมเดลที่ไม่มีการปรับค่าการเรียนรู้ขณะฝึก และในกรณีที่การปรับค่าการเรียนรู้ไม่ทำให้โมเดลมีประสิทธิภาพเพิ่มขึ้น จะหยุดการฝึกโมเดล

```

if face_module:
    input_list.append(s_model_fe1.inputs)

if nose_module:
    input_list.append(s_model_fe2.inputs)

if mouth_module:
    input_list.append(s_model_fe3.inputs)

s_model_fe = tf.keras.Model(inputs=input_list, outputs=z)

```

ภาพที่ 26 สร้างส่วนรับข้อมูลเข้าที่สามารถรับข้อมูลเข้ามากกว่าหนึ่งค่า และสร้างโมเดลข้อมูลเข้าหลายค่า

```

s_model_fe.compile(optimizer=op,
                  loss=loss_fn,
                  metrics=[train_acc_metric])

```

ภาพที่ 27 คำสั่งเพื่อตั้งค่าโมเดลให้ใช้งาน optimizer ฟังก์ชันการสูญเสีย และตั้งค่าหน่วยวัดประสิทธิภาพเป็นค่าความถูกต้อง

```

reduceLRonPlateau = tf.keras.callbacks.ReduceLRonPlateau(monitor='loss', factor = 0.8, patience=10, min_lr = 0.0000001, verbose = 1)
earlyStop = tf.keras.callbacks.EarlyStopping(patience=10, monitor='loss', verbose = 1)
callbacks_list = [reduceLRonPlateau, earlyStop]

```

ภาพที่ 28 คำสั่งเพิ่มความสามารถในขณะฝึกให้ปรับค่าการเรียนรู้ลดลงเมื่อค่าความสูญเสียคงที่ หรือหยุดการฝึก

สุดท้ายใช้คำสั่งแสดงดังภาพที่ 29 เพื่อฝึกโมเดลข้อมูลเข้าหลายค่า และใช้คำสั่งแสดงดังภาพที่ 30 เพื่อทดสอบโมเดลร่วมกับชุดข้อมูลสำหรับทดสอบ

```

hist = s_model_fe.fit(
x=tr_input,y=y_label1,
batch_size = batch_size,
#validation_data=predict_data,
epochs=200,
verbose=1,
callbacks=callbacks_list)

```

ภาพที่ 29 คำสั่งที่ใช้สำหรับฝึกโมเดล

Algorithm 2 จะเป็นขั้นตอนวิธีที่ใช้สร้างและทดสอบโมเดลข้อมูลเข้าหลายค่าและการปรับปรุงภาพ (SR) ขั้นตอนแรกจะทำตามขั้นตอนที่ 1 – 2 จะไม่แตกต่างจาก Algorithm 1 จากนั้นเพิ่มขั้นตอนการใช้ BSRGANs เพื่อปรับปรุงภาพใบหน้า และเพิ่มความคมชัดจนภาพมีขนาดมากกว่า 224 x 224 ตามขั้นตอนที่ 3 ในขั้นตอนที่ 4 จะปรับขนาดภาพใบหน้าให้มีขนาดเป็น 224 x 224 ขั้นตอนที่ 5 จะนำภาพใบหน้าไปหาภาพบริเวณจมูกและภาพบริเวณปาก โดยใช้ MTCNN จากนั้นจึงดำเนินการปรับขนาดภาพบริเวณจมูกและภาพบริเวณปากให้เป็นขนาด 224 x 224 ตามขั้นตอนที่ 6 – 7 และปรับค่าของพิกเซลให้อยู่ในช่วง 0 - 1 ในขั้นตอนที่ 8 ในขั้นตอนที่ 8 – 15 จะดำเนินการเช่นเดียวกับขั้นตอนที่ 6 -13 ของ Algorithm 1 ในส่วนของการทดสอบเริ่มต้นตั้งแต่ขั้นตอนที่ 16 ซึ่งทำซ้ำขั้นตอนที่ 1 – 9 กับชุดข้อมูลสำหรับทดสอบ และทดสอบโมเดลในขั้นตอนที่

```

def predict5(model,c_gen, file_names, step):
    time_need_to_reduce = 0
    y_p = []
    y_a = []
    i = 0
    print(step)
    while i < step:
        print(i)
        batch_x, batch_y = c_gen.__getitem__(i)
        p_result = model.predict(batch_x)
        j = 0
        if i % 10 == 9:
            start_time = time.time()
            gc.collect()
            end_time = time.time()
            time_need_to_reduce = time_need_to_reduce + end_time - start_time
        while j < len(batch_y):
            a = np.argmax(batch_y[j])

            p = np.argmax(p_result[j])
            y_a.append(a)
            y_p.append(p)
            j = j + 1
        i = i + 1

```

ภาพที่ 30 คำสั่งที่ใช้ทดสอบโมเดล

จากกระบวนการที่ได้กล่าวมาข้างต้น โมเดลที่พัฒนาตามสถาปัตยกรรมที่ออกแบบไว้ เมื่อแสดงข้อมูลชั้นและพารามิเตอร์แบบสรุปจะแสดงดังภาพที่ 31 จะเห็นว่าในชั้นข้อมูลเข้าข้อมูลเข้าที่เป็นภาพใบหน้า ภาพจมูก และภาพปาก จะไม่เชื่อมกัน และถูกดำเนินการในชั้น conv แยกจากกันจนกระทั่งถึงชั้น avg_pool จึงมีการเชื่อมต่อกันเป็นข้อมูลเข้าคุณลักษณะเข้าสู่ส่วน FC เพื่อจำแนกประเภทของภาพ

Layer (type)	Output Shape	Param #	Connected to
f_model_input_10 (InputLayer)	[(None, 224, 224, 3)]	0	
n_model_input_10 (InputLayer)	[(None, 224, 224, 3)]	0	
m_model_input_10 (InputLayer)	[(None, 224, 224, 3)]	0	
f_model_conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	f_model_input_10[0][0]
n_model_conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	n_model_input_10[0][0]
m_model_conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	m_model_input_10[0][0]
f_model_conv1_conv (Conv2D)	(None, 112, 112, 64)	9472	f_model_conv1_pad[0][0]
n_model_conv1_conv (Conv2D)	(None, 112, 112, 64)	9472	n_model_conv1_pad[0][0]
m_model_conv1_conv (Conv2D)	(None, 112, 112, 64)	9472	m_model_conv1_pad[0][0]
f_model_conv1_bn (BatchNormaliz)	(None, 112, 112, 64)	256	f_model_conv1_conv[0][0]
n_model_conv1_bn (BatchNormaliz)	(None, 112, 112, 64)	256	n_model_conv1_conv[0][0]
m_model_conv1_bn (BatchNormaliz)	(None, 112, 112, 64)	256	m_model_conv1_conv[0][0]
f_model_conv1_relu (Activation)	(None, 112, 112, 64)	0	f_model_conv1_bn[0][0]
n_model_conv1_relu (Activation)	(None, 112, 112, 64)	0	n_model_conv1_bn[0][0]
m_model_conv1_relu (Activation)	(None, 112, 112, 64)	0	m_model_conv1_bn[0][0]
.....			
f_model_conv5_block3_3_conv (Co	(None, 7, 7, 2048)	1050624	f_model_conv5_block3_2_relu[0][0]
n_model_conv5_block3_3_conv (Co	(None, 7, 7, 2048)	1050624	n_model_conv5_block3_2_relu[0][0]
m_model_conv5_block3_3_conv (Co	(None, 7, 7, 2048)	1050624	m_model_conv5_block3_2_relu[0][0]
f_model_conv5_block3_3_bn (Batac	(None, 7, 7, 2048)	8192	f_model_conv5_block3_3_conv[0][0]
n_model_conv5_block3_3_bn (Batac	(None, 7, 7, 2048)	8192	n_model_conv5_block3_3_conv[0][0]
m_model_conv5_block3_3_bn (Batac	(None, 7, 7, 2048)	8192	m_model_conv5_block3_3_conv[0][0]
f_model_conv5_block3_add (Add)	(None, 7, 7, 2048)	0	f_model_conv5_block2_out[0][0] f_model_conv5_block3_3_bn[0][0]
n_model_conv5_block3_add (Add)	(None, 7, 7, 2048)	0	n_model_conv5_block2_out[0][0] n_model_conv5_block3_3_bn[0][0]
m_model_conv5_block3_add (Add)	(None, 7, 7, 2048)	0	m_model_conv5_block2_out[0][0] m_model_conv5_block3_3_bn[0][0]
f_model_conv5_block3_out (Activ	(None, 7, 7, 2048)	0	f_model_conv5_block3_add[0][0]
n_model_conv5_block3_out (Activ	(None, 7, 7, 2048)	0	n_model_conv5_block3_add[0][0]
m_model_conv5_block3_out (Activ	(None, 7, 7, 2048)	0	m_model_conv5_block3_add[0][0]
f_model_avg_pool (GlobalAverage	(None, 2048)	0	f_model_conv5_block3_out[0][0]
n_model_avg_pool (GlobalAverage	(None, 2048)	0	n_model_conv5_block3_out[0][0]
m_model_avg_pool (GlobalAverage	(None, 2048)	0	m_model_conv5_block3_out[0][0]
concatenate (Concatenate)	(None, 6144)	0	f_model_avg_pool[0][0] n_model_avg_pool[0][0] m_model_avg_pool[0][0]
dense_17 (Dense)	(None, 3)	18435	concatenate[0][0]
Total params: 175,131,267 Trainable params: 174,676,995 Non-trainable params: 454,272			

ภาพที่ 31 สรุปรูปโครงสร้างของโมเดลข้อมูลเข้าหลายค่าที่ออกแบบ (โดยแสดงเฉพาะส่วนต้นและส่วนท้าย จากทั้งหมด 3,260 บรรทัด)

นอกจากนี้ในงานนี้ยังได้มีการพัฒนาขั้นตอนวิธีเพิ่มเติมโดยมีการใช้เทคนิค Super Resolution (SR) ปรับปรุงคุณภาพของรูปภาพก่อนนำเข้าสู่มอเดลเพื่อการจำแนกดังขั้นตอนวิธีที่ใช้ในการสร้างและทดสอบโมเดลโดยใช้ SR แสดงดัง Algorithm 2

Algorithm 2: Face-Mask Detection and Classification using Multiple Inputs and SR

Training Process

1. อ่านข้อมูลภาพ
2. ตรวจสอบบริเวณภาพใบหน้า โดยใช้ MTCNN
3. ใช้ BSRGANs ปรับปรุงภาพใบหน้าจนภาพมีขนาดมากกว่า 224×224
4. ปรับขนาดภาพใบหน้าเป็น 224×224
5. ตรวจสอบภาพบริเวณจมูกและภาพบริเวณปาก โดยใช้ MTCNN
6. ตรวจสอบบริเวณจมูกได้ภาพขนาด 50×50 และปรับขนาดเป็น 224×224
7. ตรวจสอบบริเวณปากได้ภาพขนาด 100×50 และปรับขนาดเป็น 224×224
8. ปรับช่วงค่าสีให้อยู่ระหว่าง 0 ถึง 1
9. สำหรับชุดข้อมูลสร้างข้อมูลเข้าที่ประกอบด้วย
 - กรณี 1 ภาพ คือ ภาพใบหน้า หรือ ภาพบริเวณจมูก หรือ ภาพบริเวณปาก
 - กรณี 2 ภาพ คือ ภาพใบหน้าและภาพบริเวณจมูก หรือ ภาพใบหน้าและภาพบริเวณปาก หรือ ภาพบริเวณจมูกและภาพบริเวณปาก
 - กรณี 3 ภาพ คือ ภาพใบหน้า ภาพบริเวณจมูก และ ภาพบริเวณปาก
10. สร้างโครงสร้างโมเดล ResNet152 สำหรับใบหน้า จมูก และปาก โดยกำหนดชั้นส่งออกคุณลักษณะเป็นชั้น Global Average Pooling
11. สร้างชั้นรวมคุณลักษณะใบหน้า จมูก และปากซึ่งเป็นค่าส่งออกจากชั้น Global Average Pooling
(ในตอนนี้โครงสร้างโมเดลสำหรับใบหน้า จมูก และปาก ในข้อ 10 จะถูกรวมโครงสร้างเข้าด้วยกัน)
12. สร้างชั้นแบบ Fully Connected layer ที่มีจำนวนโหนดเท่ากับจำนวนคลาส
13. สร้างชั้นนำเข้าแบบข้อมูลเข้าหลายค่าประกอบด้วยภาพใบหน้า จมูก และปาก โดยเชื่อมชั้นนำเข้าของแต่ละโมเดลในข้อ 8
14. กำหนดรูปแบบการฝึกโมเดล ได้แก่ ค่า optimizer ค่า loss และการเพิ่มความสามารถในขณะฝึก

Algorithm 2: Face-Mask Detection and Classification using Multiple Inputs and SR

15. ฝึกโมเดลโดยใช้ชุดข้อมูลฝึก จำนวน 200 รอบ

Testing Process

16. อ่านข้อมูลภาพ

17. ตรวจสอบบริเวณภาพใบหน้า โดยใช้ MTCNN

18. ใช้ BSRGANs ปรับปรุงภาพใบหน้าจนภาพมีขนาดมากกว่า 224×224

19. ปรับขนาดภาพใบหน้าเป็น 224×224

20. ตรวจสอบภาพบริเวณจมูกและภาพบริเวณปาก โดยใช้ MTCNN

21. ตรวจสอบบริเวณจมูกได้ภาพขนาด 50×50 และปรับขนาดเป็น 224×224

22. ตรวจสอบบริเวณปากได้ภาพขนาด 100×50 และปรับขนาดเป็น 224×224

23. ปรับช่วงค่าสีให้อยู่ระหว่าง 0 ถึง 1

24. สำหรับชุดข้อมูลสร้างข้อมูลเข้าที่ประกอบด้วย

กรณี 1 ภาพ คือ ภาพใบหน้า หรือ ภาพบริเวณจมูก หรือ ภาพบริเวณปาก

กรณี 2 ภาพ คือ ภาพใบหน้าและภาพบริเวณจมูก หรือ ภาพใบหน้าและภาพบริเวณปาก หรือ ภาพบริเวณจมูกและภาพบริเวณปาก

กรณี 3 ภาพ คือ ภาพใบหน้า ภาพบริเวณจมูก และ ภาพบริเวณปาก

25. ป้อนข้อมูลเข้าโมเดลเพื่อจำแนกการใส่หน้ากาก

3.2. การตรวจจับใบหน้า จมูก และปากโดยใช้ MTCNN (Face Nose Mouth Detection)

โมเดล MTCNN ถูกนำมาใช้ในการตรวจจับภาพใบหน้า ภาพบริเวณจมูกและภาพบริเวณปาก โมเดล MTCNN จะสกัดคุณลักษณะจากภาพต้นฉบับซึ่งอาจมีใบหน้ามากกว่าหนึ่งใบหน้า เช่น มีใบหน้า 3 ใบหน้าในหนึ่งภาพ แสดงดังภาพที่ 32 โดย MTCNN จะหาบริเวณที่คาดว่าจะน่าจะเป็นใบหน้า จากนั้นจึงหาตำแหน่งของตา จมูก และปาก

ตำแหน่งของตาจะได้เป็นตำแหน่งของตาซ้ายและตาขวา (สีน้ำเงิน) ตำแหน่งของจมูกจะแสดงเป็นจุดจุดเดี่ยว ตรงกลางจมูก (สีแดง) และ ตำแหน่งของปากจะเป็นตำแหน่งของริมฝีปากทั้งสองข้าง (สีเขียว) โดยมีตัวอย่างของผลลัพธ์แสดงดังภาพที่ 33

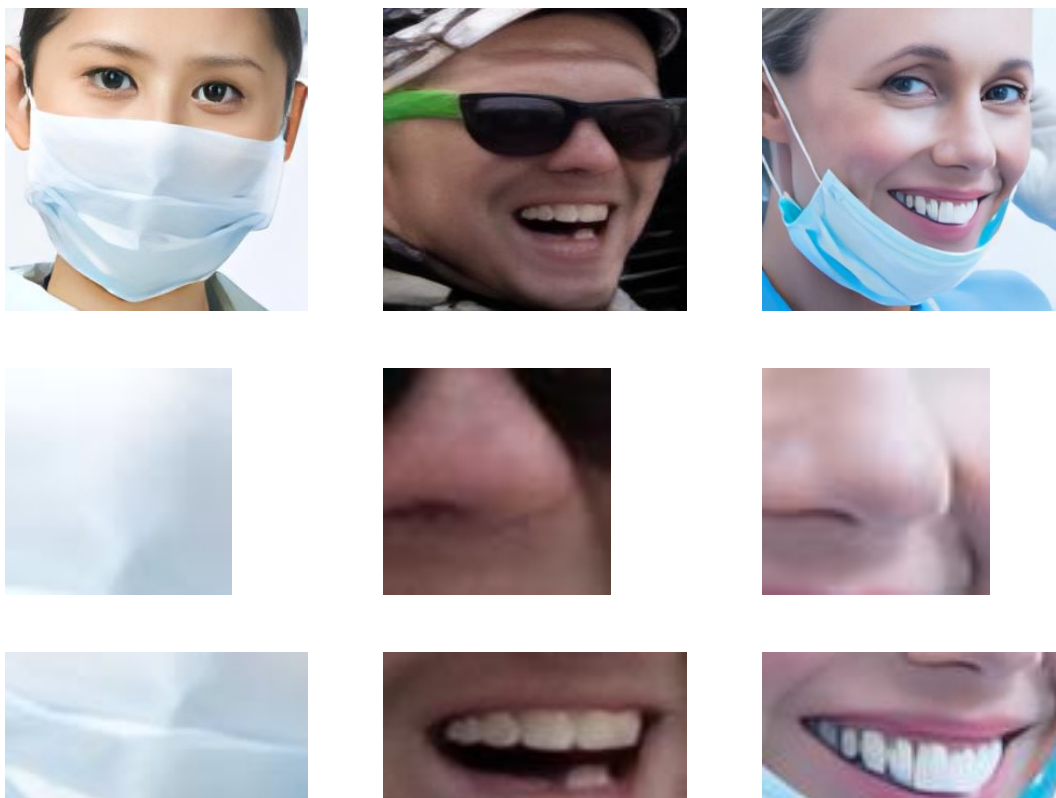


ภาพที่ 32 ตัวอย่างภาพต้นฉบับที่มีภาพใบหน้ามากกว่าหนึ่งภาพ



ภาพที่ 33 ผลลัพธ์ของ MTCNN ระบุตำแหน่งของตา จมูก และปาก

ภาพใบหน้าที่ได้จะถูกปรับขนาดให้มีขนาด 224×224 หลังจากนั้นตัดภาพบริเวณ จมูกออกมาโดยใช้ตำแหน่งของจมูกเป็นจุดศูนย์กลางและขยายออกไปเป็นภาพขนาด 50×50 ใน ส่วนของภาพปากหาตำแหน่งศูนย์กลางของตำแหน่งริมฝีปากทั้งสองข้างและขยายออกไปเป็นภาพ ขนาด 50×100 แสดงดังตัวอย่างตามภาพที่ 34 หากภาพบางภาพไม่สามารถตรวจจับจมูกและปาก ได้ จะใช้ภาพสีดำนั่นแทนที่ภาพบริเวณจมูกและปากแทน



ภาพที่ 34 ตัวอย่างภาพใบหน้า และผลลัพธ์จาก MTCNN บริเวณจมูกและปาก

การใช้งานโมเดล MTCNN เริ่มต้นจากโหลดโมเดลที่เว็บไซต์ <https://github.com/ipazc/mtcnn> จากนั้นติดตั้งตามที่คู่มือระบุ และใช้คำสั่งแสดงดั่งภาพที่ 35 เพื่อโหลดชุดคำสั่ง ตามด้วยคำสั่งแสดงดั่งภาพที่ 36 เพื่อโหลดโมเดล MTCNN และใช้คำสั่งแสดงดั่งภาพที่ 37 เพื่อตรวจจับใบหน้าจากภาพ

```
from mtcnn.mtcnn import MTCNN
```

ภาพที่ 35 โหลดชุดคำสั่งของ MTCNN

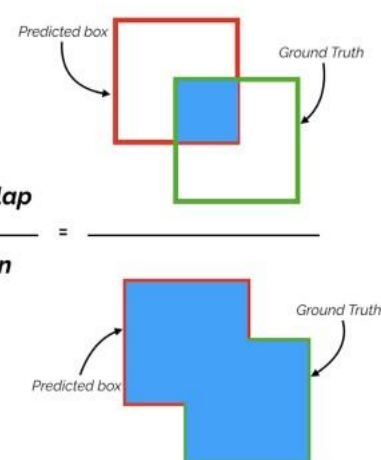
```
mtcnn_detector = MTCNN()
```

ภาพที่ 36 คำสั่งโหลดโมเดล MTCNN

```
faces = mtcnn_detector.detect_faces(t_image)
```

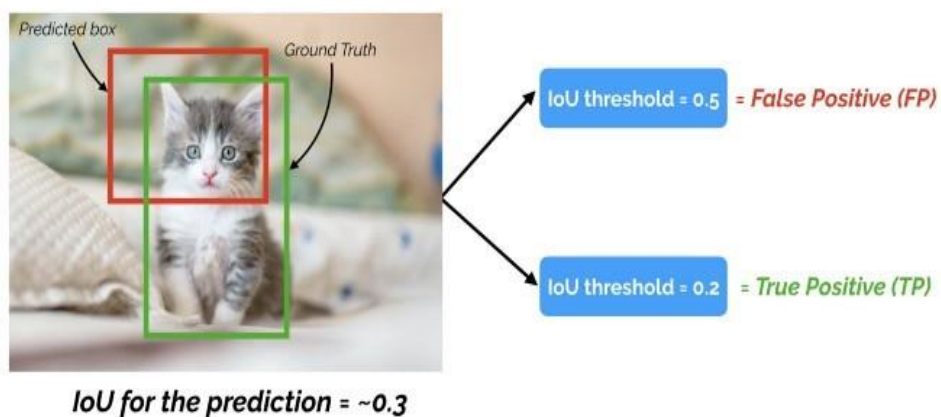
ภาพที่ 37 คำสั่งใช้โมเดลตรวจจับใบหน้าจากภาพ

ภาพใบหน้าที่ถูกตรวจจับได้ จะถูกนำไปหาค่า Intersection over Union หรือค่า IoU ซึ่งเป็นค่าหนึ่งที่ใช้ในการตัดสินผลลัพธ์ของการตรวจจับวัตถุว่าผลลัพธ์ถูกต้องหรือไม่ โดยใช้กรอบภาพ (Bounding Box) ที่ของกรอบภาพจริง (Ground Truth Bounding Box) ที่ใช้แสดงถึงตำแหน่งของวัตถุในภาพ และกรอบภาพที่จำแนก (Predict Bounding Box) ที่แสดงถึงตำแหน่งของวัตถุที่ทำนาย โดยจะใช้ค่า Intersection over Union (IoU) ที่สามารถคำนวณได้ตามภาพที่ 38 โดยหาค่าของพื้นที่ที่ทับซ้อนกันของกรอบภาพจริง และกรอบภาพที่จำแนก (Area of Overlap) หารด้วยค่าพื้นที่ของกรอบภาพจริงรวมกับกรอบภาพที่จำแนกทั้งหมด (Area of Union)

$$\text{Intersection over Union (IoU)} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{\text{Area of Overlap}}{\text{Area of Predicted box} + \text{Area of Ground Truth} - \text{Area of Overlap}}$$


ภาพที่ 38 การหา IoU [60]

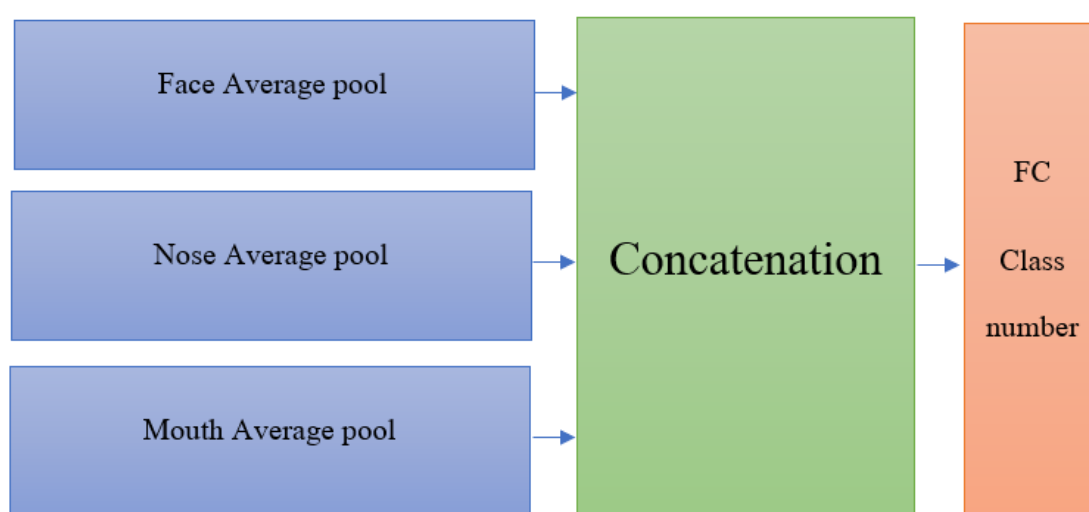
โดยค่า IoU ที่ใช้กันทั่วไป คือ 0.5 ทำให้เมื่อค่าที่ได้มากกว่าหรือเท่ากับ 0.5 จึงจะถือว่าภาพนั้น ตรวจจับถูกต้อง (TP) ถ้าน้อยกว่าหรือมีกรอบภาพซ้ำซ้อนถือว่าตรวจจับไม่ถูกต้องพลาด (FP) เมื่อไม่สามารถตรวจจับได้ หรือจำแนกประเภทไม่ถูกต้อง (FN) ภาพที่ 39 แสดงให้เห็นว่าค่า IoU ส่งผลต่อการจำแนกการตรวจจับวัตถุ (Prediction) จากตัวอย่าง IoU ที่คำนวณได้คือ 0.3



ภาพที่ 39 กรอบภาพจริง กรอบภาพที่จำแนก และการใช้งาน IoU ใน Object Detection [60]

3.3. การสกัดคุณลักษณะ (Feature Extraction) และการจำแนกประเภท (Classification)

การออกแบบการสกัดคุณลักษณะของข้อมูลเข้าหลายค่าใช้โมเดล Resnet 152 เพื่อการสกัดคุณลักษณะ โดยตัดชั้น Fully Connected (FC) ในแต่ละส่วนออก นำโมเดล Resnet 152 จำนวนสามตัวที่ผ่านการฝึกโดยใช้ข้อมูลเข้าเป็นภาพใบหน้า ภาพบริเวณจมูกและภาพบริเวณปากตามลำดับ มาประกอบกันโดยนำคุณลักษณะของทุกโมเดลย่อยมา Concatenate กัน จากนั้นค่อยส่งให้ชั้น Fully Connected (FC) ดังภาพที่ 40



ภาพที่ 40 โมเดล Resnet สามตัวมาประกอบกันโดยใช้ชั้น Concatenation

การจำแนกประเภทใช้ชั้น Fully Connected (FC) ที่ได้รับชุดตัวเลขที่เกิดจากการนำแผนที่คุณลักษณะไปผ่านชั้น Global Average Pooling ซึ่งมีจำนวนตัวเลขเท่ากับจำนวนแผนที่คุณลักษณะ ชุดตัวเลขจะถูกนำไปผ่านชั้น Concatenation ก่อน ซึ่งการ Concatenation ของ tensor จะเป็นไปตามภาพที่ 41 และตัวอย่างการ Concatenation เป็นไปตามภาพที่ 42 หลังจากนั้นส่งให้ Neural Networks (NN) ที่มีจำนวนของโหนดเท่ากับประเภทที่ต้องการจำแนก เช่น ประเภทที่ต้องจำแนกมี 3 ประเภท (3 class) ได้แก่ ประเภทใส่หน้ากาก ประเภทไม่ใส่หน้ากาก และ ประเภทใส่หน้ากากไม่ถูกต้อง

```
X= [[[1, 4, 2, 7], [7, 2, 3, 6]], [[3, 3, 5, 1], [7, 5, 7, 7]]]
Y= [[[5, 6, 6, 3], [1, 2, 3, 4]], [[3, 1, 5, 4], [7, 5, 5, 6]]]
Result= [[[1, 4, 2, 7], [7, 2, 3, 6], [5, 6, 6, 3], [1, 2, 3, 4]]
, [[3, 3, 5, 1], [7, 5, 7, 7], [3, 1, 5, 4], [7, 5, 5, 6]]]
```

ภาพที่ 41 ตัวอย่างการ Concatenation ของ Tensor

```
face= [0.09085376, 0.5146752, 0.0041788174, ...]
nose= [0.30834886, 0.0, 0.0041788174, ...]
mouth= [0.0, 0.001458348, 0.0, ...]
Result= [0.09085376, 0.30834886, 0.0, 0.5146752, 0.0, 0.001458348, ...]
```

ภาพที่ 42 ตัวอย่างการ Concatenation ชุดตัวเลข

ผู้วิจัยได้สร้าง Resnet 152 ขึ้นมาตามโมเดลใน Tensorflow ใช้คำสั่งแสดงดังภาพที่ 43 ใน การสร้างส่วน Identity Block และ Convolutional Block ของ Resnet ภาพที่ 44 นำคำสั่ง Resnetx_layer ไปประกอบกับส่วนต้นและส่วนท้ายของโมเดล Resnet เพื่อสร้างโมเดล Resnet ตั้งแต่ข้อมูลเข้าถึงข้อมูลออก ภาพที่ 45 เป็นส่วนของคำสั่ง Resnetx_main และ get_block_layer เพื่อสร้างโมเดล Resnet152

```

def Resnetx_layers(x,is_lower_than50, block_layers):
    filter_size = 64
    if len(block_layers) == 3:
        filter_size = filter_size * 2
    for i in range(len(block_layers)):

        #print('i=' + str(i))
        if i == 0:
            # For sub-block 1 Residual/Convolutional block not needed
            if is_lower_than50:
                x = convolutional_block34(x, filter_size, stride = 1)
            else:
                x = convolutional_block101(x, filter_size, stride = 1)
            for j in range(block_layers[i] - 1):
                #print('j=' + str(j))
                if is_lower_than50:
                    x = identity_block34(x, filter_size)
                else:
                    x = identity_block101(x, filter_size)
        else:
            # One Residual/Convolutional Block followed by Identity blocks
            # The filter size will go on increasing by a factor of 2
            filter_size = filter_size*2
            if is_lower_than50:
                x = convolutional_block34(x, filter_size)
            else:
                x = convolutional_block101(x, filter_size)
            for j in range(block_layers[i] - 1):
                #print('j=' + str(j))
                if is_lower_than50:
                    x = identity_block34(x, filter_size)
                else:
                    x = identity_block101(x, filter_size)

    return x

```

ภาพที่ 43 คำสั่งเพื่อสร้างส่วน Identity Block และ Convolutional Block ของ Resnet

```

def check_is_lower_than50(model_name):
    is_lower_than50 = False
    if model_name == t_34:
        is_lower_than50 = True
    return is_lower_than50

def Resnetx(model_name, shape = (224, 224, 3), block_layers = [3, 4, 6, 3], classes = 2):
    is_lower_than50 = check_is_lower_than50(model_name)
    # Step 1 (Setup Input Layer)
    x_input = tf.keras.layers.Input(shape)
    x = tf.keras.layers.ZeroPadding2D((3, 3))(x_input)
    # Step 2 (Initial Conv Layer along with maxPool)
    #conv1
    x = tf.keras.layers.Conv2D(64, kernel_size=7, strides=2, padding='valid')(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.Activation('relu')(x)
    x = tf.keras.layers.ZeroPadding2D((1, 1))(x)
    x = tf.keras.layers.MaxPool2D(pool_size=3, strides=2, padding='valid')(x)
    #conv2
    #x.shape
    # Define size of sub-blocks and initial filter size
    #block_layers = [3, 4, 23, 3]

    # Step 3 Add the Resnet Blocks
    x = Resnetx_layers(x, is_lower_than50, block_layers)
    # Step 4 End Dense Network
    x = tf.keras.layers.GlobalAveragePooling2D()(x)
    #x = tf.keras.layers.AveragePooling2D((2, 2), padding = 'same')(x)
    #x = tf.keras.layers.Flatten()(x)
    #x = tf.keras.layers.Dense(512, activation = 'relu')(x)
    x = tf.keras.layers.Dense(classes, activation = 'softmax')(x)
    model = tf.keras.models.Model(inputs = x_input, outputs = x, name = "ResNet" + model_name)
    return model

```

ภาพที่ 44 คำสั่งสร้างโมเดล Resnet เริ่มต้นแต่ข้อมูลเข้า (Input) ไปถึงข้อมูลออก (Output)

```

t_101 = "101"
t_152 = "152"
t_152m = "152m"
t_34 = "34"

def Resnetx_main(model_name, shape, class_num):
    block_layers = get_block_layers(model_name)
    return Resnetx(model_name=model_name, shape=shape, block_layers=block_layers, classes = class_num)

def get_block_layers(model_name):
    block_layers_101 = [3, 4, 23, 3]
    block_layers_152 = [3, 8, 36, 3]
    #block_layers_152m = [8, 36, 3]
    block_layers_152m = [8, 36]
    block_layers = block_layers_101
    if model_name == t_152:
        block_layers = block_layers_152
    elif model_name == t_152m:
        block_layers = block_layers_152m
    return block_layers

```

ภาพที่ 45 คำสั่งสร้างโมเดล Resnet152 และคำสั่งตั้งค่า block ของ Resnet152

3.4. การปรับปรุงภาพโดยโมเดลการเรียนรู้เชิงลึกแบบ BSRGANs

การปรับปรุงภาพโดยใช้โมเดล BSRGANs จะช่วยเพิ่มรายละเอียดของภาพใบหน้า ทำให้รายละเอียดของขอบและจุดเด่นบนใบหน้ามีความชัดเจนมากขึ้น การใช้งานโมเดล BSRGANs เริ่มต้นจากการโหลดโมเดลที่ <https://github.com/cszn/BSRGAN> หลังจากนั้นใช้โปรแกรม python ที่ชื่อ main_test_bsrgan แสดงดังภาพที่ 46 โดยจะต้องใส่ภาพในโฟลเดอร์ที่ได้ระบุไว้ในคู่มือการใช้งานโมเดลก่อน หลังจากนั้นรันโปรแกรม ผลลัพธ์จะอยู่ในโฟลเดอร์ที่ระบุตามคู่มือ ภาพที่ผ่านการใช้งานโมเดล BSRGANs จะเป็นภาพที่มีขนาดเล็กกว่า 224 x 224 และในกรณีที่ภาพผลลัพธ์ยังมีขนาดเล็กกว่า 224 x 224 จะใช้งานโมเดล BSRGANs ซ้ำ ตัวอย่างการทำงานภาพจากชุดข้อมูล FMLD ที่ใช้โมเดล BSRGANs แสดงได้ดังภาพที่ 47

```
def image_enhancement_sr():  
    %cd BSRGAN  
    !python main_test_bsrgan.py  
    %cd ..
```

ภาพที่ 46 คำสั่งเพื่อใช้งานโมเดล BSRGANs



ภาพที่ 47 ตัวอย่างภาพต้นฉบับ (บน) และภาพที่ผ่านการใช้งาน BSRGANs (ล่าง)

บทที่ 4

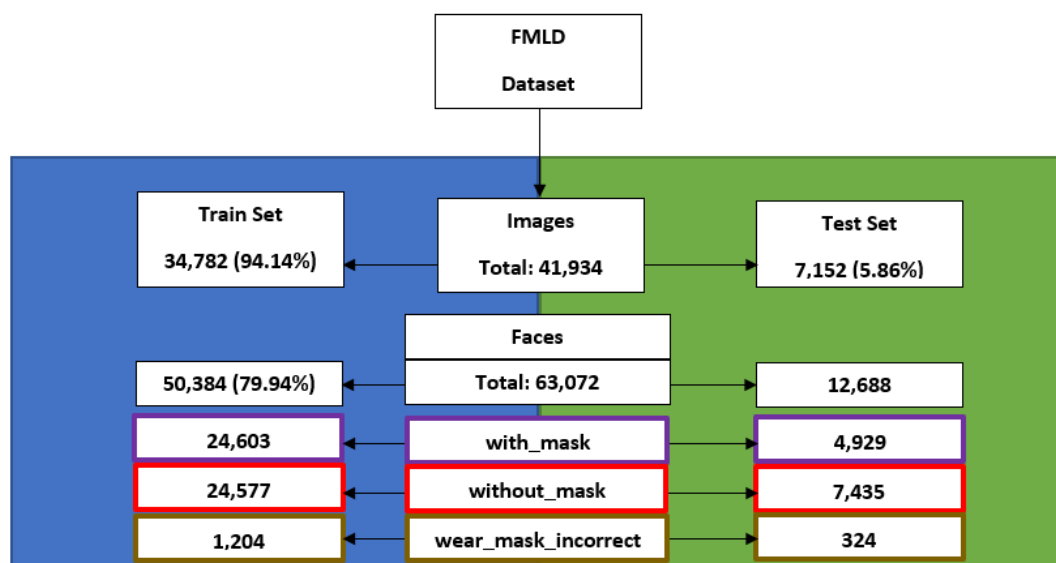
การทดลองและผลการทดลอง

บทนี้จะกล่าวถึงชุดข้อมูลที่ใช้ในการทดลอง แสดงผลการทดลองตามที่ได้ออกแบบและพัฒนาขั้นตอนวิธีที่สถาปัตยกรรมของโมเดลข้อมูลเข้าหลายค่าที่ได้กล่าวไป เพื่อแสดงให้เห็นถึงประสิทธิภาพของโมเดลข้อมูลเข้าหลายค่าเทียบกับโมเดลข้อมูลเข้าเดี่ยว โดยมีประเด็นต่าง ๆ ในการทดลองดังนี้ ประเด็นแรก แสดงประสิทธิภาพโมเดลที่รับข้อมูลเข้าหลายค่าเปรียบเทียบกับโมเดลที่รับข้อมูลเข้าเดี่ยว ประเด็นที่สอง แสดงประสิทธิภาพของโมเดลเมื่อมีการปรับปรุงภาพมาแก้ปัญหาภาพที่ไม่ชัดเจนด้วยเทคนิค Super-Resolution (SR) โดยใช้โมเดล BSRGANs ทั้งสองประเด็นใช้ค่าความถูกต้อง (Accuracy), ค่าความเที่ยงตรง (Precision), ค่าความแม่นยำ (Recall) และค่า F1 (F1-score) เป็นตัววัดประสิทธิภาพของโมเดล

4.1. ชุดข้อมูลที่ใช้ในการทดลอง

ชุดข้อมูลใช้เป็นข้อมูลหลัก 2 ชุด ชุดที่ 1 คือ FMLD และชุดที่ 2 AFMDK ชุดข้อมูลที่มาจกชุดข้อมูลมาตรฐาน FMLD เป็นชุดข้อมูลที่ผสมระหว่างชุดข้อมูล MAFA [61] และชุดข้อมูล Wider Face [62] โดยชุดข้อมูล FMLD มีรายละเอียดแสดงดังภาพที่ 48 มีภาพต้นฉบับทั้งหมด 41,934 ภาพ เนื่องจากบางภาพอาจมีหลายใบหน้าทำให้มีภาพใบหน้า 63,072 ใบหน้า แบ่งเป็น 2 ชุดข้อมูล ได้แก่ ชุดข้อมูลสำหรับฝึก (Train) และชุดข้อมูลสำหรับทดสอบ (Test) และประเภทของใบหน้า 3 ประเภทคือประเภทสวมหน้ากาก ประเภทไม่สวมหน้ากาก และประเภทสวมหน้ากากไม่ถูกต้อง โดยมีรายละเอียดดังนี้

- ชุดข้อมูลสำหรับฝึก (Train) มีภาพต้นฉบับทั้งหมด 34,782 ภาพ มี Annotation เป็นภาพใบหน้า 50,384 ใบหน้า แบ่งเป็นประเภทสวมหน้ากาก (with_mask) 24,603 ภาพ ประเภทไม่สวมหน้ากาก (without_mask) 24,577 ภาพ และประเภทสวมหน้ากากไม่ถูกต้อง (wear_mask_incorrect) 1,204 ภาพ
- ชุดข้อมูลสำหรับทดสอบ (Test) มีภาพต้นฉบับทั้งหมด 7,152 ภาพ มี Annotation เป็นภาพใบหน้า 12,688 ใบหน้า แบ่งเป็นประเภทสวมหน้ากาก (with_mask) 4,929 ภาพ ประเภทไม่สวมหน้ากาก (without_mask) 7,435 ภาพ และประเภทสวมหน้ากากไม่ถูกต้อง (wear_mask_incorrect) 324 ภาพ

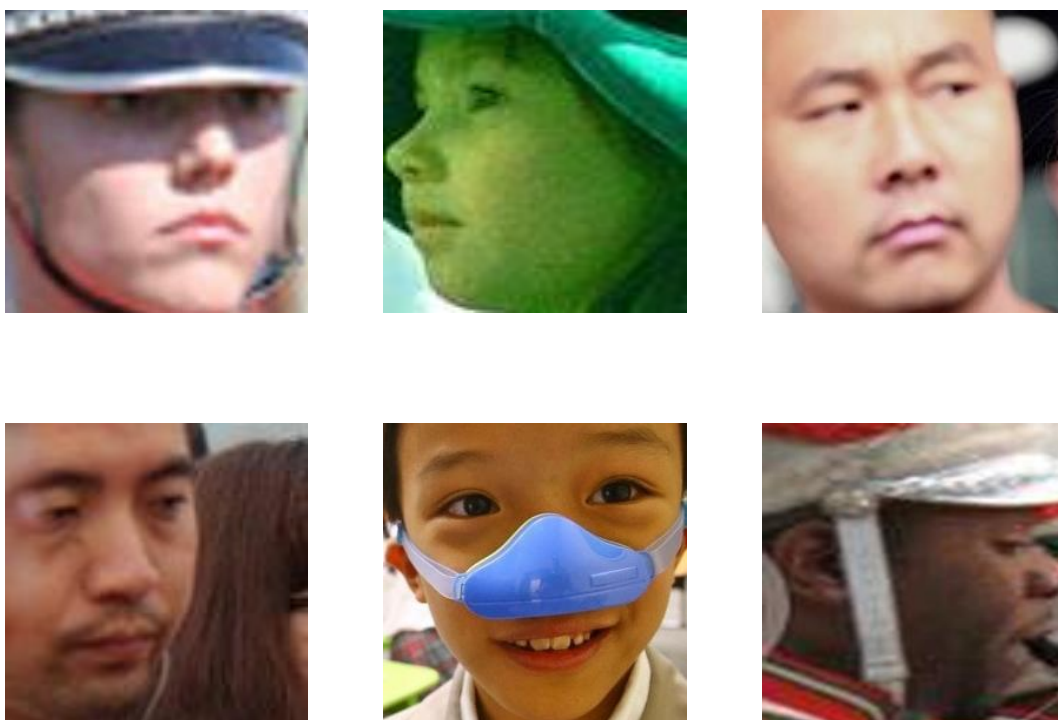


ภาพที่ 48 แสดงจำนวนภาพต้นฉบับ ภาพใบหน้าของชุดข้อมูล FMLD แบ่งตามประเภทของใบหน้า และประเภทของชุดข้อมูลฝึกและทดสอบ

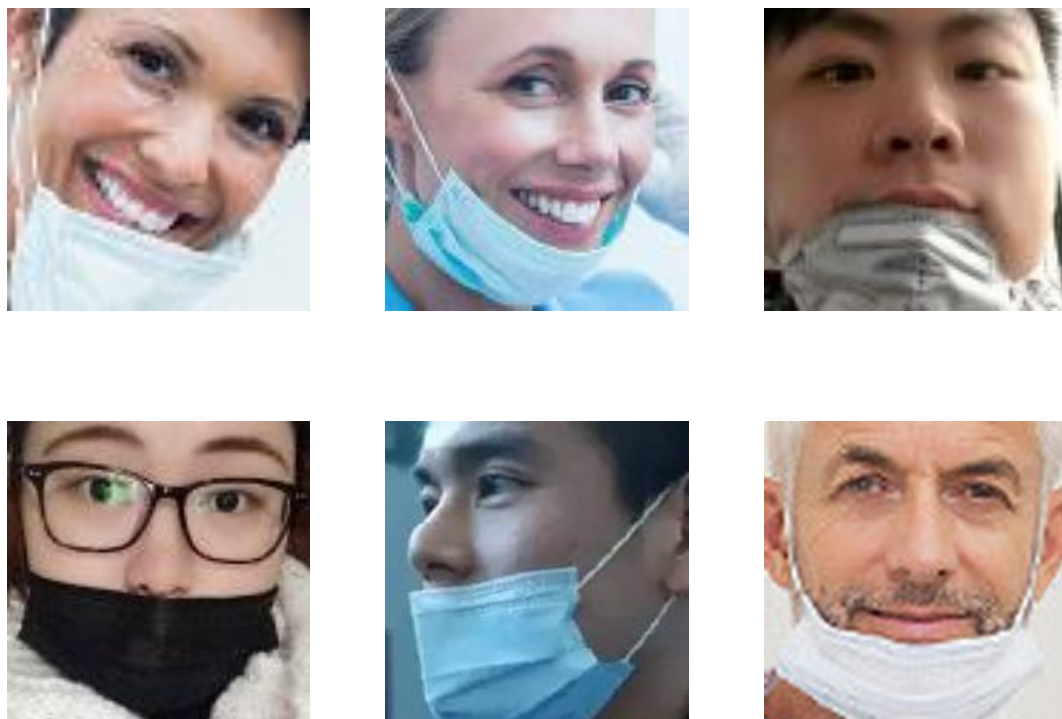
ภาพตัวอย่างจากชุดข้อมูล FMLD ภาพใบหน้าสวมหน้ากากถูกต้อง (with_mask) แสดงดังภาพที่ 49 ภาพใบหน้าประเภทไม่สวมหน้ากาก (without_mask) แสดงดังภาพที่ 50 และ ภาพใบหน้าประเภทสวมหน้ากากไม่ถูกต้อง (mask_wear_incorrect) แสดงดังภาพที่ 51 นอกจากนี้ ชุดข้อมูลนี้ก็มีภาพที่มีความยากในการจำแนกประเภทสูง เช่น ภาพมีขนาดเล็ก ภาพไม่ชัด และภาพที่มีจุดจางไม่เห็นรายละเอียดแสดงดังภาพที่ 52 เป็นต้น



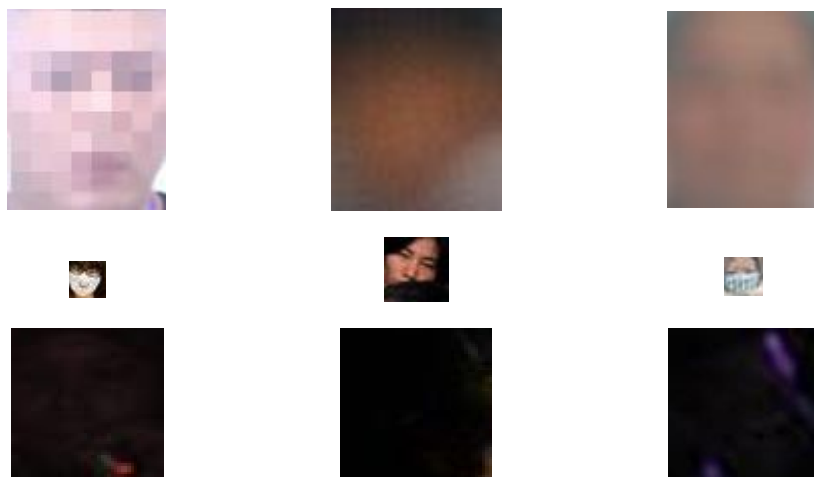
ภาพที่ 49 ตัวอย่างชุดข้อมูล FMLD ประเภทสวมหน้ากากถูกต้อง



ภาพที่ 50 ตัวอย่างชุดข้อมูล FMLD ประเภทไม่สวมหน้ากาก



ภาพที่ 51 ตัวอย่างชุดข้อมูล FMLD ประเภทสวมหน้ากากไม่ถูกต้อง



ภาพที่ 52 ตัวอย่างชุดข้อมูล FMLD แถวบนสุดภาพไม่ชัด แถวกลางภาพเล็ก และแถวล่างสุดภาพมืด

ชุดข้อมูลที่ 2 คือ ชุดข้อมูล AFMDK มีรายละเอียด คือ มีภาพต้นฉบับทั้งหมด 853 ภาพ ซึ่งทั้งหมดมีใบหน้า 4,072 ใบหน้า เป็นชุดข้อมูลต้นฉบับไม่ได้มีการแยกชุดข้อมูลสำหรับฝึกและชุดข้อมูลสำหรับทดสอบ แบ่งเป็นประเภทสวมหน้ากาก (with_mask) 3,232 ภาพ ประเภทไม่สวมหน้ากาก (without_mask) 717 ภาพและประเภทสวมหน้ากากไม่ถูกต้อง (wear_mask_incorrect) 123 ภาพ แสดงดังตารางที่ 8

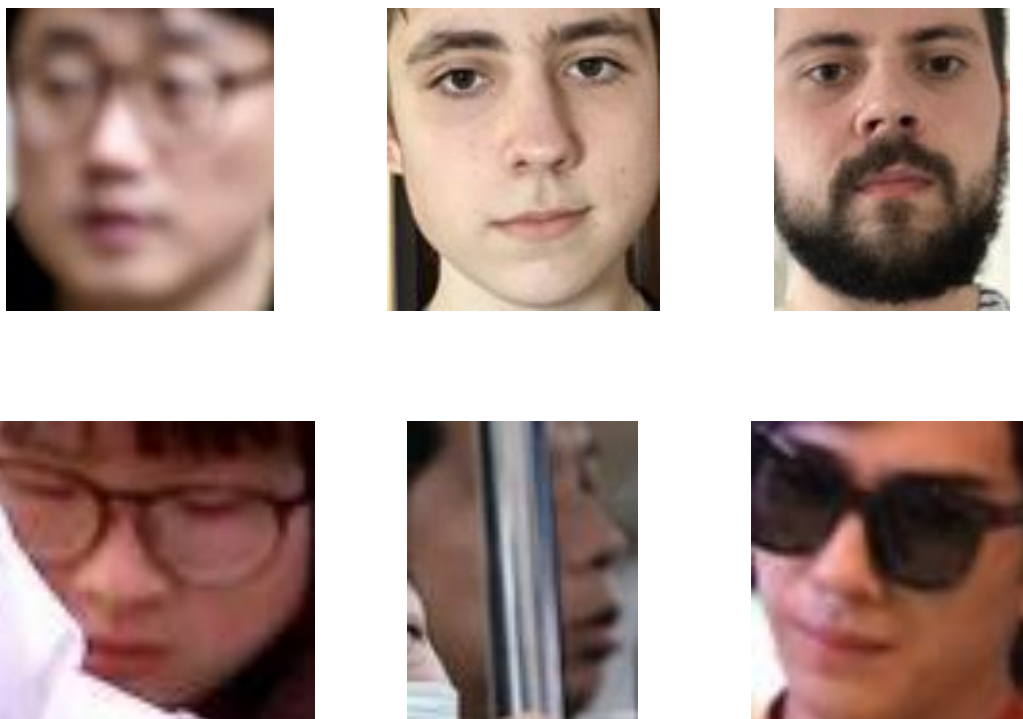
ภาพตัวอย่างจากชุดข้อมูล AFMDK ได้แก่ 1) ภาพใบหน้าสวมหน้ากากถูกต้อง (with_mask) แสดงดังภาพที่ 53 2) ประเภทไม่สวมหน้ากาก (without_mask) แสดงดังภาพที่ 54 และ 3) ประเภทสวมหน้ากากไม่ถูกต้อง (mask_wear_incorrect) แสดงดังภาพที่ 55

ตารางที่ 8 แสดงจำนวนภาพใบหน้าแบ่งตามประเภทของภาพใบหน้าของชุดข้อมูล AFMDK

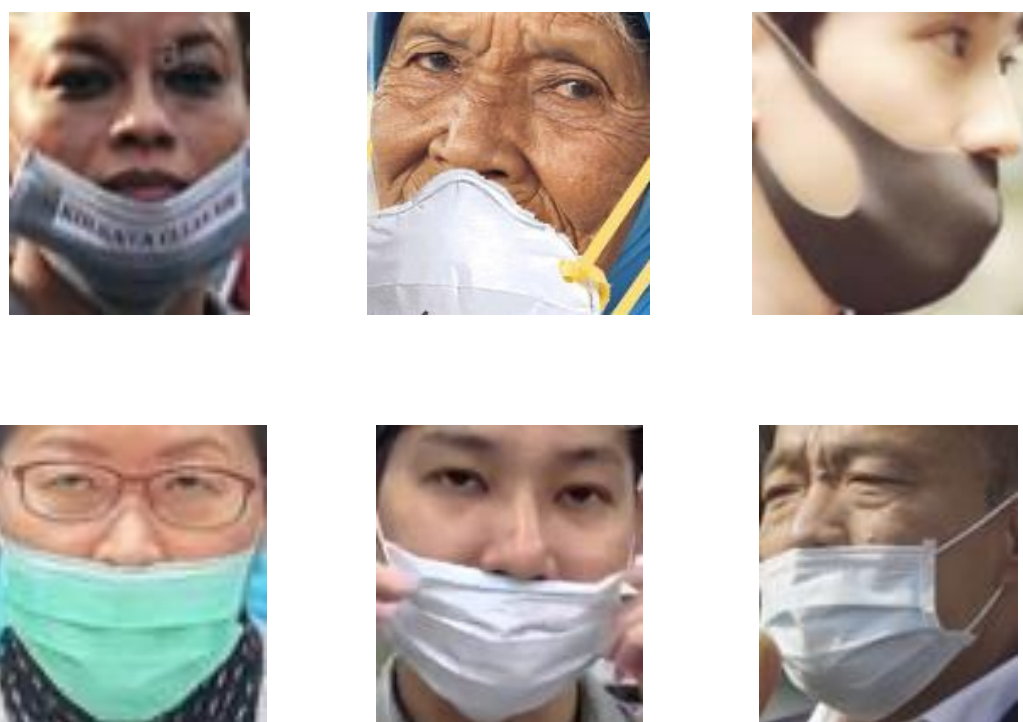
Class	with_mask	without_mask	wear_mask_incorrect	Total
Face Images	3,232 (79%)	717 (18%)	123 (3%)	4,072 (100%)



ภาพที่ 53 ตัวอย่างชุดข้อมูล AFMDK ประเภทสวมหน้ากากถูกต้อง



ภาพที่ 54 ตัวอย่างชุดข้อมูล AFMDK ประเภทไม่สวมหน้ากาก



ภาพที่ 55 ตัวอย่างชุดข้อมูล AFMDK ประเภทสวมหน้ากากไม่ถูกต้อง

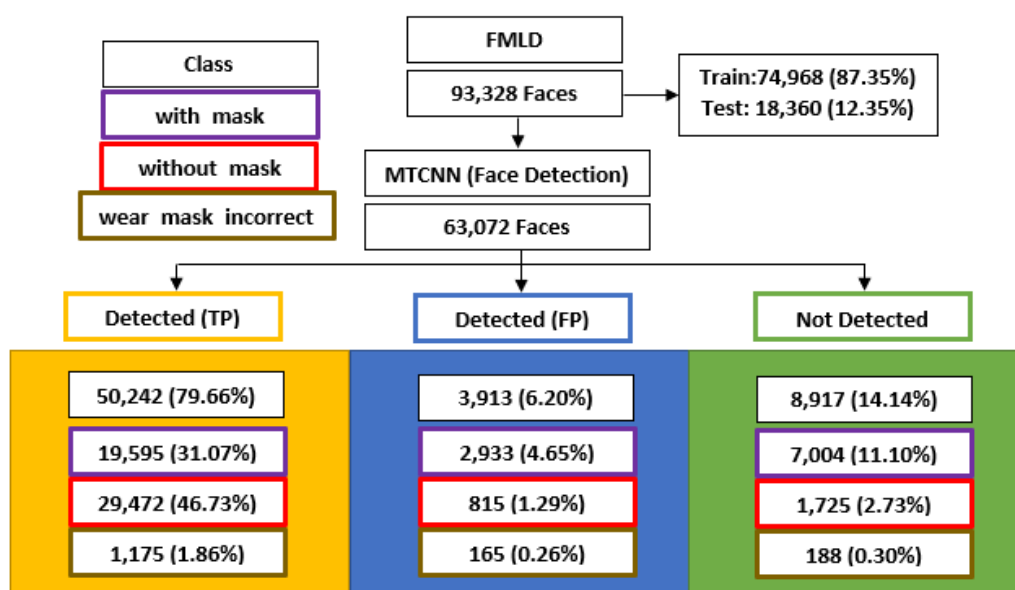
4.2. ประสิทธิภาพการตรวจจับใบหน้าโดยใช้ MTCNN

ในหัวข้อนี้วิเคราะห์ผลการตรวจจับใบหน้าโดยใช้ MTCNN บนชุดข้อมูล FMLD แบ่งออกเป็นชุดข้อมูลสำหรับฝึก และชุดข้อมูลสำหรับทดสอบตามประเภทภาพต้นฉบับ โดยมีภาพใบหน้าในชุดข้อมูลทั้งหมด 93,328 ใบหน้า แบ่งเป็นชุดข้อมูลสำหรับฝึก 74,968 ใบหน้า ชุดข้อมูลสำหรับทดสอบ 18,360 ใบหน้า มีรายละเอียดแสดงดังภาพที่ 56 โดยภาพใบหน้าที่ได้จากการตรวจจับใบหน้าผ่าน MTCNN จะถูกแบ่งออกเป็นกลุ่ม ได้แก่

Detected (TP) ภาพใบหน้าที่ในกลุ่มนี้จะเป็นภาพใบหน้าที่ถูกตรวจจับได้ที่มีอยู่ใน Annotation บนชุดข้อมูลต้นฉบับและมีค่า IoU มากกว่าหรือเท่ากับ 0.5 มีทั้งหมด 50,242 ภาพ แบ่งเป็นประเภทใส่หน้ากากถูกต้อง 19,595 ภาพ ประเภทไม่ใส่หน้ากาก 29,472 ภาพ และประเภทใส่หน้ากากไม่ถูกต้อง 1,175 ภาพ ซึ่งกลุ่มนี้จะถูกนำไปสร้างเป็นชุดข้อมูลสำหรับการทดลองโดยให้ชื่อชุดข้อมูล FMLD MTCNN

Detected (FP) ภาพใบหน้าที่ตรวจจับได้ แต่ค่า IoU ต่ำกว่า 0.5 หรือกล่าวได้ว่าตรวจจับได้แต่ผิด มีทั้งหมด 3,913 ภาพ แบ่งเป็นประเภทใส่หน้ากากถูกต้อง 2,933 ภาพ ประเภทไม่ใส่หน้ากาก 815 ภาพ และประเภทใส่หน้ากากไม่ถูกต้อง 165 ภาพ

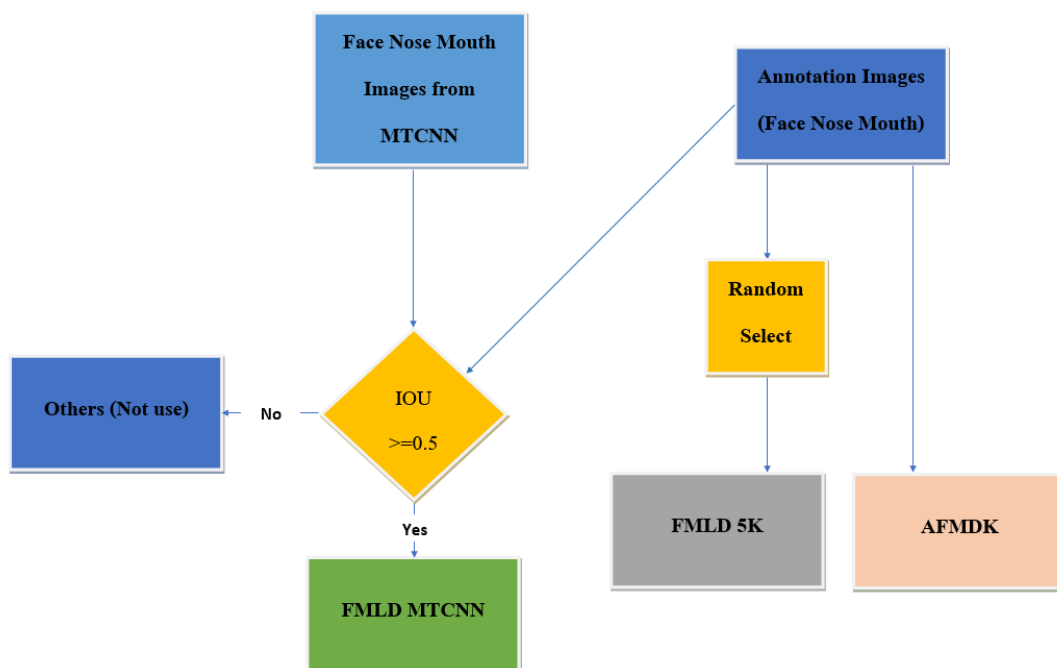
Not Detected ภาพใบหน้าที่มีอยู่ใน Annotation ของข้อมูลต้นฉบับแต่ MTCNN ตรวจจับไม่ได้ มี 8,917 ภาพ แบ่งเป็นประเภทใส่หน้ากากถูกต้อง 7,004 ภาพ ประเภทไม่ใส่หน้ากาก 1,725 ภาพ และประเภทใส่หน้ากากไม่ถูกต้อง 188 ภาพ



ภาพที่ 56 แสดงรายละเอียดของภาพใบหน้าในชุดข้อมูล FMLD MTCNN ของกลุ่ม Detected (TP), Detected (FP) และ Not Detected

4.3. การเตรียมชุดข้อมูลที่ใช้ในการทดลองโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่า

ในหัวข้อนี้จะกล่าวถึงการเตรียมชุดข้อมูลที่ใช้ทำการทดลองจำนวน 3 ชุดข้อมูล ได้แก่ FMLD MTCNN, FMLD 5K และ AFMDK โดยในแต่ละชุดข้อมูลมีการเตรียมภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก ดังรายละเอียดต่อไปนี้ 1) FMLD MTCNN คือ ชุดข้อมูลที่เกิดจากการนำภาพต้นฉบับของชุดข้อมูล FMLD ไปตรวจจับภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปากโดยใช้ MTCNN และวัดค่า IOU หากมีค่ามากกว่าหรือเท่ากับ 0.5 จะนำภาพที่ตรวจจับได้ถูกต้องมาสร้างเป็นชุดข้อมูล แต่ในชุดข้อมูล 2) FMLD 5K และ 3) AFMDK จะเป็นภาพใบหน้าจาก Annotation ที่นำมาผ่าน MTCNN เพื่อหาภาพบริเวณจมูก และภาพบริเวณปาก ขั้นตอนการในการสร้างชุดข้อมูล FMLD MTCNN, FMLD 5K และ AFMDK แสดงดังภาพที่ 57 ในส่วนของชุดข้อมูล 2 class ทดสอบโดยการผสมประเภทสวมหน้ากากไม่ถูกต้อง และประเภทไม่สวมหน้ากากเป็นประเภทเดียว

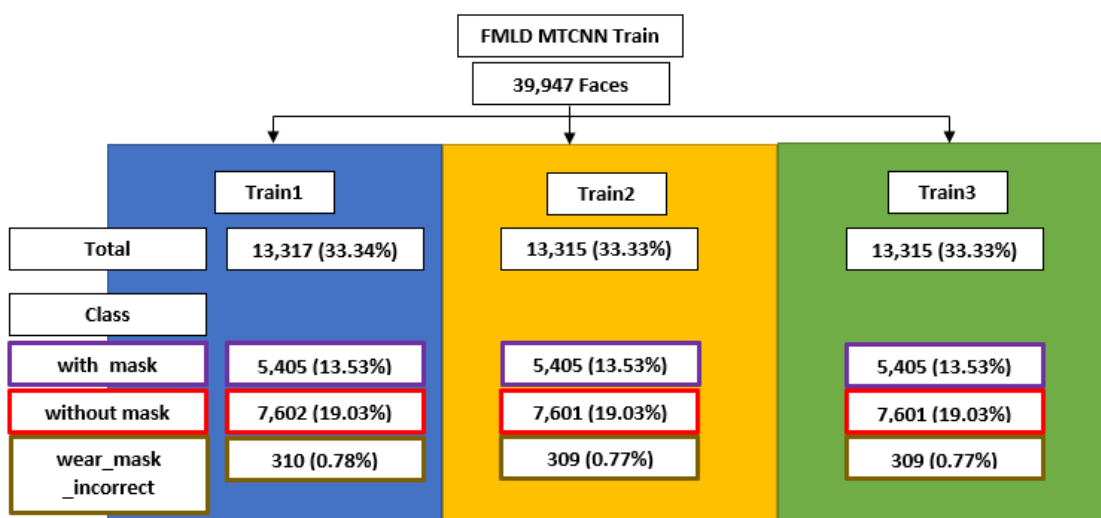


ภาพที่ 57 แสดงขั้นตอนการได้มาซึ่งชุดข้อมูล FMLD MTCNN, FMLD 5K, AFMDK

เนื่องจากมีข้อจำกัดในเรื่องของประสิทธิภาพเครื่องคอมพิวเตอร์ที่ไว้ใช้ในการทดลองชุดข้อมูล FMLD MTCNN มีการแบ่งภาพใบหน้าบัพผิออกออกเป็น 3 ส่วน เท่า ๆ กัน โดยภาพในแต่ละ

ส่วนได้จากการสุ่มภาพจากชุดข้อมูลสำหรับฝึกและชุดข้อมูลสำหรับทดสอบตามชุดข้อมูลต้นฉบับโดยให้อัตราส่วนระหว่างประเภทแต่ละประเภทคงเดิม เพื่อใช้ในการฝึกโมเดลโดยใช้ชุดข้อมูลสำหรับการฝึกที่ได้แบ่งออกเป็นสามให้ชื่อโมเดลตามแต่ละส่วนคือโมเดล 1 โมเดล 2 และโมเดล 3 โดยมีรายละเอียดดังนี้

- ชุดข้อมูลสำหรับฝึก (Train) จะมีใบหน้าทั้งหมด 13,317 13,315 13,315 ภาพตามลำดับแต่ละส่วน แบ่งเป็นประเภทสวมหน้ากากถูกต้อง 5,405 ภาพในทุกส่วน ประเภทไม่สวมหน้ากาก 7,602 7,601 7,601 ภาพ ตามลำดับแต่ละส่วน และประเภทสวมหน้ากากไม่ถูกต้อง 310 309 309 ภาพตามลำดับแต่ละส่วนแสดงดังภาพที่ 58
- ชุดข้อมูลสำหรับทดสอบ (Test) ใช้ชุดข้อมูลทดสอบชุดเดียวกันซึ่งมีภาพใบหน้าจำนวนทั้งหมด 10,295 ภาพ แบ่งเป็นประเภทสวมหน้ากากถูกต้อง 3,380 ภาพ ประเภทไม่สวมหน้ากาก 6,668 ภาพ และประเภทสวมหน้ากากไม่ถูกต้อง 247 ภาพ จะได้ว่าชุดข้อมูล FMLD MTCNN มีชุดข้อมูลสำหรับฝึกและชุดข้อมูลสำหรับทดสอบแสดงดังตารางที่ 9



ภาพที่ 58 แสดงจำนวนภาพใบหน้าของแต่ละส่วนของชุดข้อมูลฝึกที่ถูกแบ่งออกเป็นส่วนของชุดข้อมูล FMLD MTCNN แบ่งตามประเภทของใบหน้า

ตารางที่ 9 แสดงจำนวนภาพใบหน้าแบ่งตามประเภทของชุดข้อมูลและประเภทของภาพใบหน้าของชุดข้อมูล FMLD MTCNN

Dataset	with_mask	without_mask	wear_mask_incorrect	Total Image
Train1	5,405 (11%)	7,602 (15%)	310 (0.6%)	13,317 (26.6%)
Train2	5,405 (11%)	7,601 (15%)	309 (0.5%)	13,315 (26.5%)
Train3	5,405 (11%)	7,601 (15%)	309 (0.5%)	13,315 (26.5%)
Test	3,380 (6%)	6,668 (14%)	247 (0.4%)	10,295 (20.4%)
Total	19,595 (39%)	29,472 (59%)	1,175 (2%)	50,242 (100%)

ชุดข้อมูล FMLD 5K มีการแบ่งภาพใบหน้าเป็นชุดข้อมูลสำหรับฝึก และชุดข้อมูลสำหรับทดสอบ โดยภาพในแต่ละประเภทของชุดข้อมูลได้จากการสุ่มภาพจากชุดข้อมูลสำหรับฝึกและชุดข้อมูลสำหรับทดสอบตามชุดข้อมูลต้นฉบับ โดยให้อัตราส่วนของชุดข้อมูลสำหรับฝึก และชุดข้อมูลสำหรับทดสอบเป็น 70 : 30 และให้แต่ละประเภทมีจำนวนใกล้เคียงกันที่สุด เนื่องจากภาพใบหน้าประเภทสวมหน้ากากไม่ถูกต้องมีอยู่น้อย จึงให้จำนวนภาพใบหน้าประเภทนี้รวมกับประเภทไม่ใส่หน้ากากรวมกันแล้วใกล้เคียงกับจำนวนภาพใบหน้าประเภทสวมหน้ากากถูกต้อง ชุดข้อมูลมีภาพใบหน้าทั้งหมด 5,000 ภาพ โดยมีรายละเอียดแสดงดังตารางที่ 10 คือ

- ชุดข้อมูลสำหรับฝึก (Train) มีภาพใบหน้าทั้งหมด 3,500 ภาพ แบ่งเป็นประเภทสวมหน้ากากถูกต้อง 1,750 ภาพ ประเภทไม่สวมหน้ากาก 875 ภาพ และประเภทสวมหน้ากากไม่ถูกต้อง 875 ภาพ
- ชุดข้อมูลสำหรับทดสอบ (Test) มีภาพใบหน้าจำนวนทั้งหมด 1,500 ภาพ แบ่งเป็นประเภทสวมหน้ากากถูกต้อง 750 ภาพ ประเภทไม่สวมหน้ากาก 423 ภาพ และประเภทสวมหน้ากากไม่ถูกต้อง 327 ภาพ

ตารางที่ 10 แสดงจำนวนภาพใบหน้าแบ่งตามประเภทของชุดข้อมูลและประเภทของภาพใบหน้าของชุดข้อมูล FMLD 5K

Dataset	with_mask	without_mask	wear_mask_incorrect	Total Image
Train	1,750 (35%)	875 (17.5%)	875 (17.5%)	3,500 (70%)
Test	750 (15%)	423 (8.5%)	327 (6.5%)	1,500 (30%)
Total	2,500 (50.00%)	1,298 (26%)	1,202 (24%)	5,000 (100%)

ชุดข้อมูล AFMDK มีการแบ่งภาพใบหน้าเป็นชุดข้อมูลสำหรับฝึก และชุดข้อมูลสำหรับทดสอบ เนื่องจากชุดข้อมูลต้นฉบับไม่ได้มีการแบ่งชุดข้อมูลสำหรับฝึกและชุดข้อมูลสำหรับทดสอบ จึงได้มีการสุ่มภาพมาสร้างเป็นชุดข้อมูลสำหรับฝึกและชุดข้อมูลสำหรับทดสอบในอัตราส่วน 70:30 ชุดข้อมูลมีภาพใบหน้าทั้งหมด 4,072 ภาพ โดยมีรายละเอียดแสดงดังตารางที่ 11 คือ

- ชุดข้อมูลสำหรับฝึก (Train) มีภาพใบหน้าทั้งหมด 2,850 ภาพ แบ่งเป็นประเภทสวมหน้ากากถูกต้อง 2,250 ภาพ ประเภทไม่สวมหน้ากาก 511 ภาพ และประเภทสวมหน้ากากไม่ถูกต้อง 89 ภาพ
- ชุดข้อมูลสำหรับทดสอบ (Test) มีภาพใบหน้าจำนวนทั้งหมด 1,222 ภาพ แบ่งเป็นประเภทสวมหน้ากากถูกต้อง 982 ภาพ ประเภทไม่สวมหน้ากาก 206 ภาพ และประเภทสวมหน้ากากไม่ถูกต้อง 34 ภาพ

ตารางที่ 11 แสดงจำนวนภาพใบหน้าแบ่งตามประเภทของชุดข้อมูลและประเภทของภาพใบหน้าของชุดข้อมูล AFMDK

Dataset	with_mask	without_mask	wear_mask_incorrect	Total Image
Train	2,250 (55%)	511 (13%)	89 (2%)	2,850 (70%)
Test	982 (24%)	206 (5%)	34 (1%)	1,222 (30%)
Total	3,232(79%)	717 (18%)	123 (3%)	4,072 (100%)

4.4. การกำหนดพารามิเตอร์การฝึกโมเดลและทดสอบโมเดลข้อมูลเข้าเดี่ยวและข้อมูลเข้าหลายค่า

การฝึกโมเดลใช้ภาพในชุดข้อมูล FMLD MTCNN, FMLD 5K และ AFMDK ที่ได้จากโมเดล MTCNN ที่มีภาพใบหน้า ภาพบริเวณจมูกและภาพบริเวณปากเป็นข้อมูลเข้า โดยกรณีที่โมเดลเป็นโมเดลข้อมูลเข้าเดี่ยว (Single Input) จะเลือกประเภทภาพใบหน้า ภาพบริเวณจมูกและภาพบริเวณปากเพียงประเภทเดียวมาฝึก และใช้ประเภทนั้นในการทดสอบด้วย โมเดลข้อมูลเข้าหลายค่า (Multiple Input) จะเลือกประเภทภาพใบหน้า ภาพบริเวณจมูกและภาพบริเวณปากมากกว่าหนึ่งภาพในการฝึก และการทดสอบ มีการกำหนดพารามิเตอร์ในการฝึกโมเดล ดังนี้

- Epoch 200 Epoch
- learning rate (lr) 0.001 และมีการปรับค่าลดลงเมื่อค่าความสูญเสียถึงจุดอิ่มตัว
- optimizer Adam และมีการหยุดการฝึกก่อนกำหนดเมื่อค่าความสูญเสียไม่มีการเปลี่ยนแปลง

ในการทำการทดลองใช้เครื่องคอมพิวเตอร์ที่มีการ์ดจอ RTX 3060 12GB, RTX 3080Ti 12GB พบว่าในส่วนของชุดข้อมูล FMLD MTCNN โมเดลข้อมูลเข้าเดี่ยวจะใช้เวลาในการฝึกประมาณ 42,665.4 วินาที หรือประมาณ 11 ชั่วโมง และโมเดลข้อมูลเข้าหลายค่าจะใช้เวลาในการฝึกประมาณ 123,456.3 วินาที หรือประมาณ 33 ชั่วโมง

4.5. การทดลองโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่า

การเรียนรู้เชิงลึกมีความสามารถในการรับข้อมูลเข้ามากกว่าหนึ่งค่า และข้อมูลเข้าก็ไม่จำเป็นต้องเป็นชนิดเดียวกันอาจเป็นข้อมูลภาพหรือข้อมูลอื่น ๆ เช่น ข้อมูลตัวอักษร ตัวเลข เป็นต้น การตรวจจับใบหน้าที่สวมหน้ากากมีประเภทอยู่ 3 ประเภท ได้แก่ ประเภทใส่หน้ากากถูกต้อง ซึ่งหน้ากากจะปิดบริเวณจมูกและปากทั้งหมด ประเภทไม่ใส่หน้ากาก ที่จะเปิดให้เห็นบริเวณจมูกและปากในกรณีทั่วไป หรือส่วนน้อยที่มีอะไรมาปิดไว้ แต่ไม่ใช่หน้ากาก เช่น มือ สิ่งของต่าง ๆ เป็นต้น และประเภทสุดท้าย ประเภทใส่หน้ากากไม่ถูกต้องที่ในภาพจะมีหน้ากากอยู่ แต่หน้ากากไม่ได้ปิดบริเวณจมูก หรือปากทั้งหมด ทำให้เห็นจมูก หรือ ปากอย่างใดอย่างหนึ่ง หรือทั้งคู่ จึงได้เกิดแนวคิดที่จะส่งภาพบริเวณจมูก และปากให้กับโมเดลเป็นข้อมูลเข้านอกเหนือจากข้อมูลเข้าภาพใบหน้า โดยให้โมเดลข้อมูลเข้าเดี่ยว คือ โมเดลตรวจจับการสวมหน้ากากที่ใช้ข้อมูลเข้าเป็นภาพเดี่ยว ได้แก่ ภาพใบหน้า ภาพจมูก และภาพปาก และโมเดลข้อมูลเข้าหลายค่า คือ โมเดลการตรวจจับการสวมหน้ากากที่ใช้ข้อมูลเข้าเป็นภาพมากกว่าหนึ่งภาพ โดยเลือกภาพมาจากภาพใบหน้า ภาพบริเวณจมูก และภาพ

บริเวณปาก ทำการทดลองโดยใช้ชุดข้อมูล 3 ชุดข้อมูล ได้แก่ FMLD 5K, FMLD MTCNN และ AFMDK ซึ่งถูกจัดเตรียมไว้ตามหลักการที่ได้กล่าวไว้ในหัวข้อที่ 4.3. ฝึกโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่าโดยมีขั้นตอนการทดลอง ได้แก่ 1) ฝึกโมเดลและทดสอบโมเดล 2) ประเมินประสิทธิภาพของโมเดล

มีการสร้างโมเดลแบ่งเป็น 2 กรณี คือ

1) การสร้างโมเดลเพื่อการจำแนกใบหน้าสวมหน้ากากแบ่งออกเป็น 2 ประเภท คือ ประเภทใส่หน้ากากถูกต้องและประเภทไม่ใส่หน้ากากรวมกับประเภทใส่หน้ากากไม่ถูกต้อง

2) การสร้างโมเดลเพื่อการจำแนกใบหน้าสวมหน้ากากแบ่งออกเป็น 3 ประเภท คือ ประเภทใส่หน้ากากถูกต้อง ประเภทไม่ใส่หน้ากากและประเภทใส่หน้ากากไม่ถูกต้อง

การประเมินประสิทธิภาพของโมเดลใช้ค่าความถูกต้อง (Accuracy), ค่าความเที่ยงตรง (Precision), ค่าความแม่นยำ (Recall) และค่า F1 (F1-score) เป็นตัววัดประสิทธิภาพของโมเดล ต่อไปจะเป็นการแสดงผลการเปรียบเทียบประสิทธิภาพของโมเดลที่ใช้ข้อมูลเข้าแบบเดี่ยวและข้อมูลเข้าแบบหลายค่า แบ่งตามชุดข้อมูล ได้แก่ FMLD 5K, FMLD MTCNN และ AFMDK และแบ่งตามประเภทของชุดข้อมูล ได้แก่ กรณี 2 ประเภท และ 3 ประเภท

4.5.1. เปรียบเทียบประสิทธิภาพของโมเดลที่ใช้ข้อมูลเข้าแบบเดี่ยวและข้อมูลเข้าหลายค่า

บนชุดข้อมูล FMLD 5K

จากผลการทดลองบนชุดข้อมูล FMLD 5K กรณีจำแนกภาพใบหน้าออกเป็น 2 ประเภท (2 class) แสดงดังตารางที่ 12 และกรณีจำแนกภาพใบหน้าเป็น 3 ประเภท (3 class) แสดงดังตารางที่ 13 จะเห็นได้ว่าโมเดลข้อมูลเข้าหลายค่าจะมีประสิทธิภาพสูงกว่าโมเดลข้อมูลเข้าเดี่ยว ทั้งในกรณีชุดข้อมูล 2 ประเภท และ 3 ประเภท

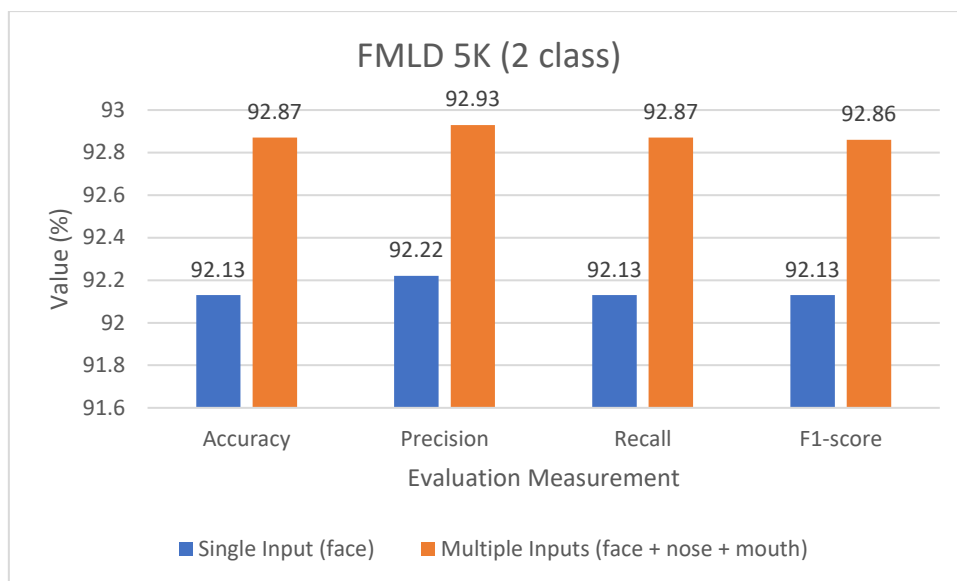
ตารางที่ 12 ผลการทดลองบนชุดข้อมูล FMLD 5K (2 class) ของโมเดลข้อมูลเข้าเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input)

Dataset	Evaluation Measurement (%)	Single Input			Multiple Inputs			
		face	nose	mouth	nose + mouth	face + nose	face + mouth	face + nose + mouth
FMLD 5K (2 class)	Accuracy	92.13	81.40	75.40	81.87	92.67	92.67	92.87
	Precision	92.22	84.33	79.09	84.25	92.74	92.74	92.93
	Recall	92.13	81.40	75.40	81.87	92.67	92.67	92.87
	F1-score	92.13	80.99	74.60	81.55	92.66	92.66	92.86

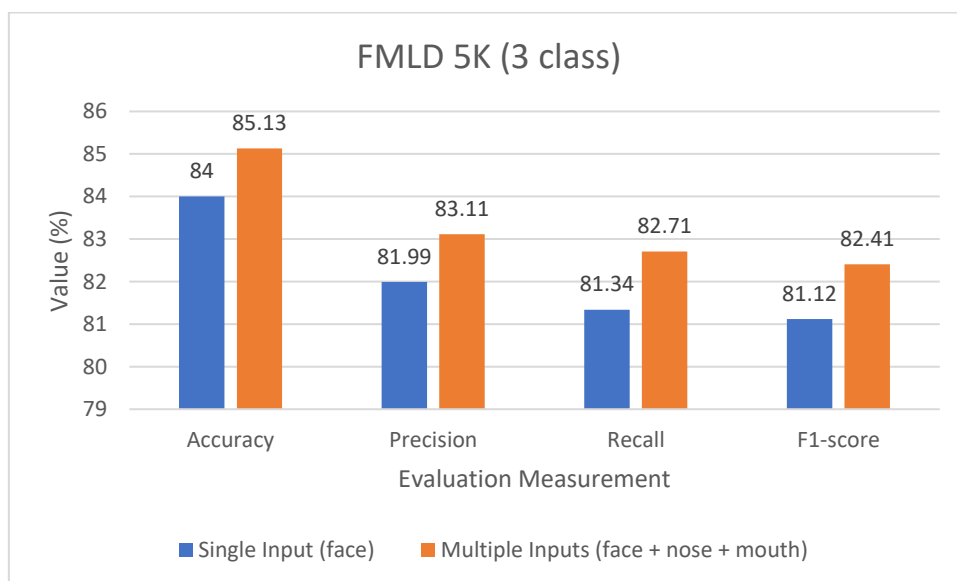
ตารางที่ 13 ผลการทดลองบนชุดข้อมูล FMLD 5K (3 class) โดยใช้ข้อมูลเข้าเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input)

Dataset	Evaluation Measurement (%)	Single Input			Multiple Inputs			
		face	nose	mouth	nose + mouth	face + nose	face + mouth	face + nose + mouth
FMLD 5K (3 class)	Accuracy	84.00	68.27	67.67	71.80	85.00	84.73	85.13
	Precision	81.99	63.47	66.59	69.94	82.94	82.75	83.11
	Recall	81.34	58.95	59.92	63.22	82.43	82.18	82.71
	F1-score	81.12	59.54	61.29	64.69	82.16	81.97	82.41

เมื่อพิจารณาผลการทดลองโดยสรุปบนชุดข้อมูล FMLD 5K จะพบว่าโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่าจะพบว่าการผสมรวมภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก (face + nose + mouth) ให้ผลดีที่สุดในกรณีการจำแนกเป็น 2 ประเภทและ 3 ประเภท แสดงดังภาพที่ 59 และภาพที่ 60 ตามลำดับ



ภาพที่ 59 กราฟแสดงประสิทธิภาพของชุดข้อมูล FMLD 5K (2 class) ของโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่าที่มีประสิทธิภาพสูงสุด



ภาพที่ 60 กราฟแสดงประสิทธิภาพของชุดข้อมูล FMLD 5K (3 class) ของโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่าที่มีประสิทธิภาพสูงสุด

4.5.2. เปรียบเทียบประสิทธิภาพของโมเดลที่ใช้ข้อมูลเข้าแบบเดี่ยวและข้อมูลเข้าหลายค่า บนชุดข้อมูล FMLD MTCNN

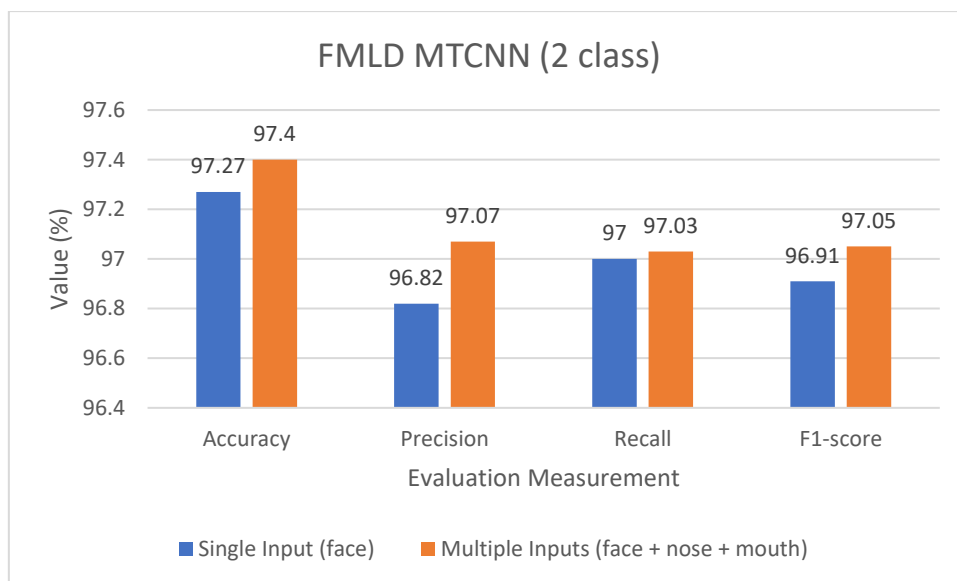
ในการทดลองบนชุดข้อมูล FMLD MTCNN มีการสร้างโมเดลจากชุดข้อมูลฝึก 3 ชุด คือ Train1, Train2 และ Train3 โดยให้ชื่อโมเดลตามชุดข้อมูลฝึก คือ โมเดล 1 โมเดล 2 และโมเดล 3 แล้วทำการวัดผลจากทั้ง 3 โมเดลที่ได้ จากการทดลองบนชุดข้อมูล FMLD MTCNN กรณีจำแนกเป็น 2 ประเภท (2 class) และกรณีจำแนกเป็น 3 ประเภท (3 class) แสดงดังตารางที่ 14 และตารางที่ 15 ตามลำดับ จะเห็นว่าโดยภาพรวมในกรณีจำแนกเป็น 2 ประเภท โมเดลข้อมูลเข้าหลายค่า (face + nose + mouth) มีประสิทธิภาพดีกว่าโมเดลข้อมูลเข้าเดี่ยว (face) สำหรับกรณีจำแนกเป็น 3 ประเภทจะเห็นว่าประสิทธิภาพใกล้เคียงกันเนื่องจากมีภาพที่ใช้ในการฝึกเพียงพอ ทำให้ประสิทธิภาพของโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่าไม่แตกต่างกันมาก ประสิทธิภาพโดยเฉลี่ยของของโมเดลที่ใช้ข้อมูลเข้าเดี่ยวและข้อมูลเข้าหลายค่าสำหรับกรณีจำแนกเป็น 2 ประเภทและกรณีจำแนกเป็น 3 ประเภทแสดงดังภาพที่ 61 และภาพที่ 62 ตามลำดับ

ตารางที่ 14 ผลการทดลองบนชุดข้อมูล FMLD MTCNN (2 class) ของโมเดลข้อมูลเข้าเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า

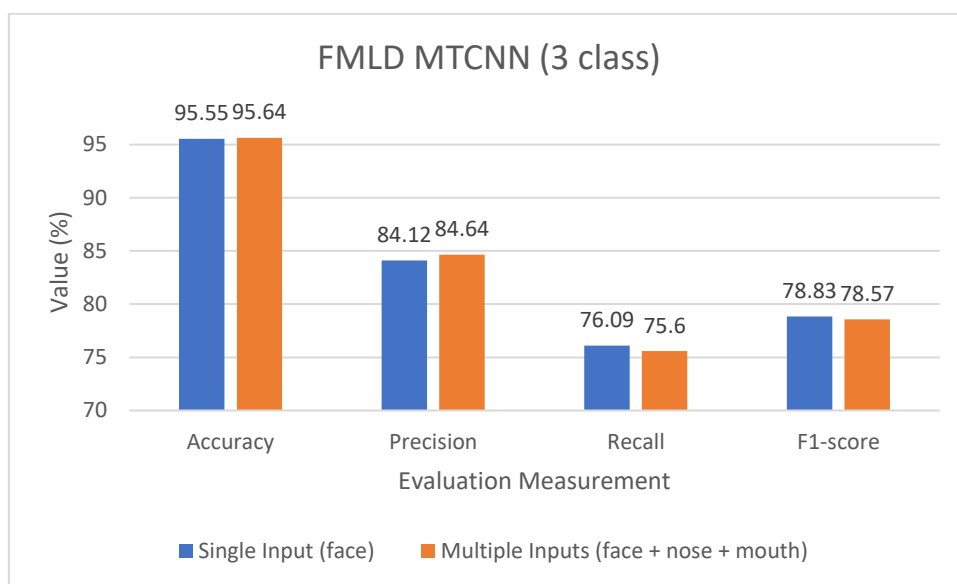
Dataset	Evaluation Measurement (%/s)	Model	Single Input	Multiple Inputs
			Face	face + nose + mouth
FMLD MTCNN (2 class)	Accuracy	โมเดล 1	97.12	97.46
		โมเดล 2	97.20	97.74
		โมเดล 3	97.48	97.00
		Average	97.27	97.40
	Precision	โมเดล 1	96.68	97.14
		โมเดล 2	96.65	97.40
		โมเดล 3	97.12	96.67
		Average	96.82	97.07
	Recall	โมเดล 1	96.79	97.11
		โมเดล 2	97.05	97.48
		โมเดล 3	97.17	96.51
		Average	97.00	97.03
	F1-score	โมเดล 1	96.73	97.13
		โมเดล 2	96.84	97.44
		โมเดล 3	97.15	96.59
		Average	96.91	97.05

ตารางที่ 15 ผลการทดลองบนชุดข้อมูล FMLD MTCNN (3 class) ของโมเดลข้อมูลเข้าเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input)

Dataset	Evaluation Measurement (%/s)	Model	Single Input	Multiple Inputs
			Face	face + nose + mouth
FMLD MTCNN (3 class)	Accuracy	โมเดล 1	95.83	95.98
		โมเดล 2	95.36	95.65
		โมเดล 3	95.45	95.29
		Average	95.55	95.64
	Precision	โมเดล 1	85.53	86.92
		โมเดล 2	82.91	84.86
		โมเดล 3	83.93	82.13
		Average	84.12	84.64
	Recall	โมเดล 1	75.60	76.30
		โมเดล 2	76.06	74.47
		โมเดล 3	76.62	76.03
		Average	76.09	75.60
	F1-score	โมเดล 1	78.85	79.72
		โมเดล 2	78.45	77.66
		โมเดล 3	79.20	78.33
		Average	78.83	78.57



ภาพที่ 61 กราฟแสดงประสิทธิภาพเฉลี่ย 3 โมเดลของชุดข้อมูล FMLD MTCNN (2 class) ของโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่า



ภาพที่ 62 กราฟแสดงประสิทธิภาพเฉลี่ย 3 โมเดลของชุดข้อมูล FMLD MTCNN (3 class) ของโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่า

4.5.3. เปรียบเทียบประสิทธิภาพของโมเดลที่ใช้ข้อมูลเข้าแบบเดี่ยวและข้อมูลเข้าหลายค่า บนชุดข้อมูล AFMDK

ชุดข้อมูล AFMDK มีภาพเรียบง่ายกว่าเมื่อเทียบกับชุดข้อมูล FMLD เนื่องจากภาพใบหน้าสวมหน้ากากถูกต้อง ภาพใบหน้าไม่สวมหน้ากาก และภาพใบหน้าสวมหน้ากากไม่ถูกต้องมีข้อแตกต่างชัดเจนกว่า ภาพใบหน้าไม่สวมหน้ากากไม่มีอะไรมาปิดบังใบหน้าทำให้ประสิทธิภาพของโมเดลข้อมูลเข้าเดี่ยวและข้อมูลเข้าหลายค่าไม่แตกต่างกันมากนัก จากผลการทดลองบนชุดข้อมูล AFMDK กรณีจำแนกออกเป็น 2 ประเภท (2 class) และกรณีจำแนกออกเป็น 3 ประเภท (3 class) แสดงถึงผลการทดลองตามตารางที่ 16 และตารางที่ 17 ตามลำดับ จะเห็นได้ว่าโมเดลข้อมูลเข้าหลายค่ามีประสิทธิภาพดีกว่าโมเดลข้อมูลเข้าเดี่ยว โดยกรณีจำแนกเป็น 2 ประเภท พบว่าเมื่อใช้ชุดข้อมูลเข้าหลายค่าแบบผสมรวมภาพใบหน้าและภาพบริเวณจมูก (face + nose) และแบบผสมรวมภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก (face + nose + mouth) มีประสิทธิภาพดีที่สุด ส่วนกรณีจำแนกเป็น 3 ประเภท พบว่าเมื่อใช้ข้อมูลเข้าแบบผสมรวมภาพใบหน้าและภาพบริเวณจมูก (face + nose) มีประสิทธิภาพดีที่สุด

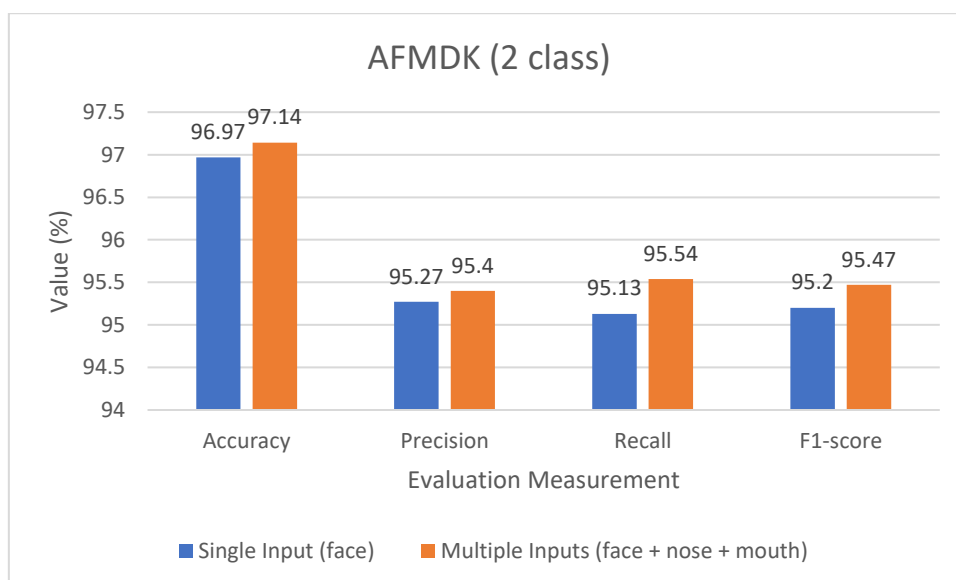
ตารางที่ 16 ผลการทดลองชุดข้อมูล AFMDK (2 class) โดยใช้โมเดลข้อมูลเข้าแบบเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input)

Dataset	Evaluation Measurement (%)	Single Input			Multiple Inputs			
		face	nose	mouth	nose + mouth	face + nose	face + mouth	face + nose + mouth
AFMDK (2 class)	Accuracy	96.97	92.47	91.24	92.96	97.14	96.97	97.14
	Precision	95.27	93.59	90.07	94.4	95.40	95.14	95.40
	Recall	95.13	82.09	80.86	83.03	95.54	95.28	95.54
	F1-score	95.20	86.36	84.41	87.30	95.47	95.21	95.47

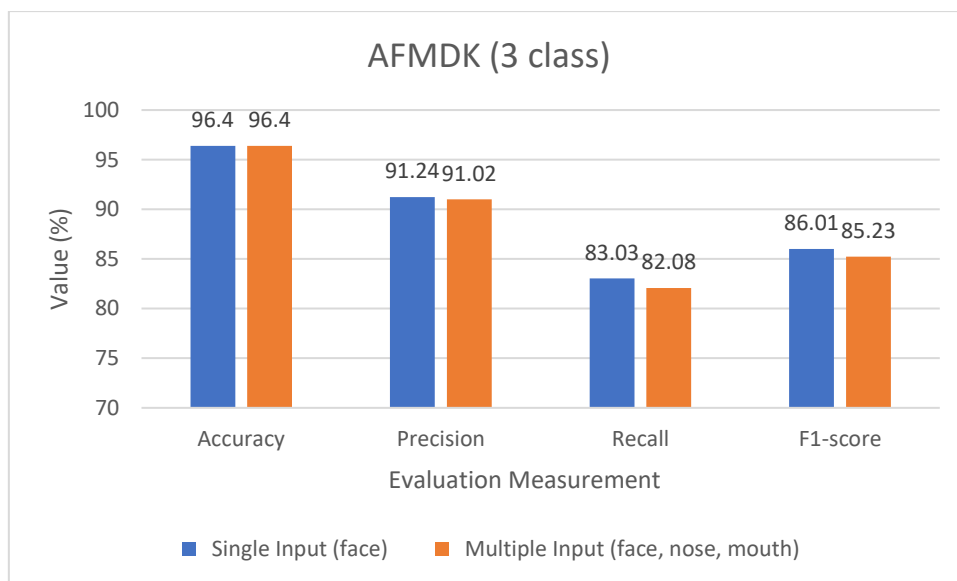
ตารางที่ 17 ผลการทดลองชุดข้อมูล AFMDK (3 class) โดยใช้โมเดลข้อมูลเข้าแบบเดี่ยว (Single Input) และโมเดลข้อมูลเข้าหลายค่า (Multiple Input)

Dataset	Evaluation Measurement (%)	Single Input			Multiple Inputs			
		face	nose	mouth	nose + mouth	face + nose	face + mouth	face + nose + mouth
AFMDK (3 class)	Accuracy	96.40	91.41	90.43	91.57	96.48	96.24	96.40
	Precision	91.24	75.67	64.64	79.12	91.38	90.70	91.02
	Recall	83.03	61.24	55.33	62.38	83.06	80.94	82.08
	F1-score	86.01	65.80	58.20	67.63	86.10	84.28	85.23

จากนั้นได้นำผลลัพธ์ของที่ดีที่สุดเมื่อใช้ข้อมูลเข้าเดี่ยวและผลลัพธ์โมเดลข้อมูลเข้าหลายค่าที่ใช้ภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปากเป็นข้อมูลเข้าที่ดีที่สุดตามผลการทดลองของส่วนใหญ่มาแสดงเปรียบเทียบกัน ในกรณีจำแนกเป็น 2 ประเภท (2 class) และกรณีจำแนกเป็น 3 ประเภท (3 class) และกราฟแสดงดังภาพที่ 63 และภาพที่ 64 ตามลำดับ



ภาพที่ 63 กราฟแสดงประสิทธิภาพของชุดข้อมูล AFMDK (2 class) ของโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่าที่มีประสิทธิภาพสูงสุด



ภาพที่ 64 กราฟแสดงประสิทธิภาพของชุดข้อมูล AFMDK (3 class) ของโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่า

4.6. การเพิ่มประสิทธิภาพของโมเดลโดยใช้การปรับปรุงภาพ

การทดลองใช้การปรับปรุงภาพ (Image Enhancement) โดยใช้ SR ในชุดข้อมูล FMLD และชุดข้อมูล AFMDK นอกจากนี้ยังได้ทำการวิเคราะห์ผลการทดสอบตามขนาดของภาพ ซึ่งได้ทำการสำรวจขนาดของภาพในแต่ละชุดข้อมูลแสดงสัดส่วนของจำนวนภาพขนาดเล็กกว่า 50×50 พิกเซลและสัดส่วนของจำนวนภาพขนาดเล็กกว่า 30×30 พิกเซลสำหรับชุดข้อมูล FMLD 5K และ AFMDK แสดงดังตารางที่ 18 และตารางที่ 19 ตามลำดับ

ตารางที่ 18 แสดงรายละเอียดของชุดข้อมูล FMLD 5K แบ่งเป็นชุดข้อมูลทั้งหมด ชุดข้อมูลสำหรับฝึก และชุดข้อมูลสำหรับทดสอบ

Dataset	FMLD 5K	FMLD 5K 50 พิกเซล	FMLD 5K 30 พิกเซล
Total	5,000 (100%)	580 (11.60%)	8 (0.16%)
Train	3,500 (70%)	320 (6.40%)	1 (0.02%)
Test	1,500 (30%)	260 (5.20%)	7 (0.14%)

ตารางที่ 19 แสดงรายละเอียดของชุดข้อมูล AFMDK แบ่งเป็นชุดข้อมูลทั้งหมด ชุดข้อมูลสำหรับฝึก และชุดข้อมูลสำหรับทดสอบ

Dataset	AFMDK	AFMDK 50 พิกเซล	AFMDK 30 พิกเซล
Total	4,072 (100%)	3,412 (83.79%)	2,671 (65.59%)
Train	2,850 (70%)	2,386 (58.60%)	1,866 (45.82%)
Test	1,222 (30%)	1,026 (25.19%)	805 (19.77%)

จากข้อมูลดังกล่าวจะเห็นว่าจำนวนภาพขนาดเล็กกว่า 30 x 30 พิกเซล มีจำนวนน้อยเกินไปในชุดข้อมูล FMLD 5K จึงไม่เหมาะแก่การทำการวิเคราะห์ ดังนั้นในการทดลองนี้จึงนิยามภาพขนาดเล็กด้วยภาพที่มีขนาดเล็กกว่า 50 x 50 พิกเซล หรือ 2,500 พิกเซล และเขียนกำกับชุดข้อมูลด้วย size < 50 ส่วนภาพที่มีขนาดมากกว่าหรือเท่ากับ 2,500 พิกเซลจะเขียนกำกับด้วย size >= 50 ผลการปรับปรุงภาพจะแสดงในหัวข้อต่อไป

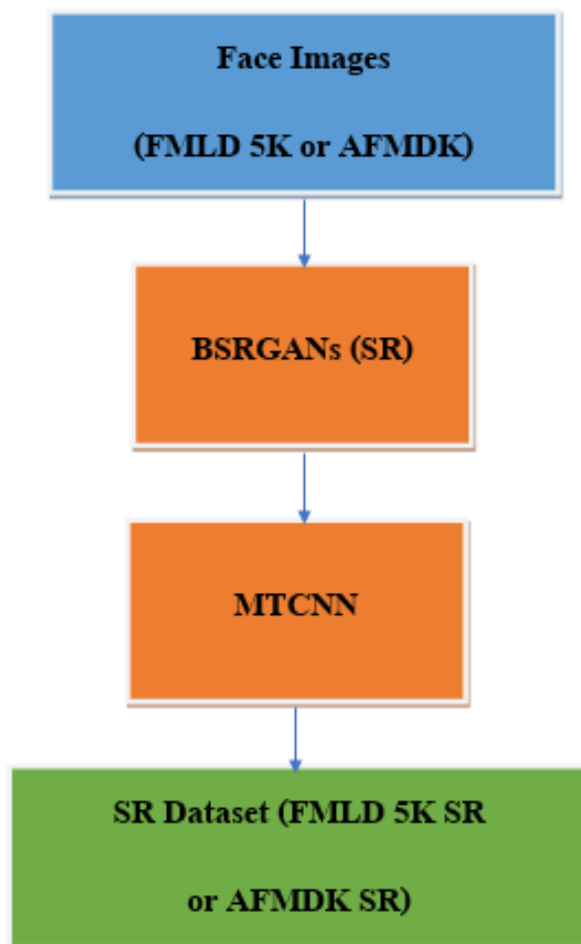
4.6.1. การปรับปรุงภาพโดยใช้ Super-Resolution (SR)

ในการทดลองนี้มีการนำชุดข้อมูลที่ได้กล่าวไปแล้วในการทดลองก่อนหน้านี้ คือ ชุดข้อมูล FMLD 5K และชุดข้อมูล AFMDK มาใช้เพื่อศึกษาผลของการปรับปรุงภาพที่มีต่อโมเดลจำแนกภาพใบหน้าที่สวมหน้ากาก โดยจะมีขั้นตอนในการสร้างชุดข้อมูลแสดงดิงภาพที่ 65 ที่ภาพใบหน้าที่ทั้งสองชุดข้อมูล จะถูกนำไปเป็นข้อมูลเข้าให้โมเดล BSRGANs เพื่อปรับปรุงภาพ ซึ่งเป็นเทคนิค Super-Resolution (SR) หลังจากนั้นจึงนำไปหาภาพบริเวณจมูกและภาพบริเวณปาก

ในหัวข้อนี้จะกำหนดชื่อเรียกให้กับชุดข้อมูล FMLD 5K และ AFMDK ตามการปรับปรุงภาพและขนาดภาพดังนี้

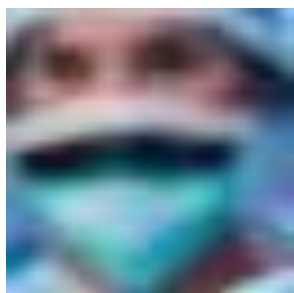
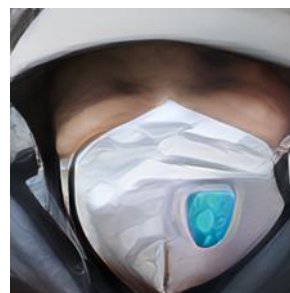
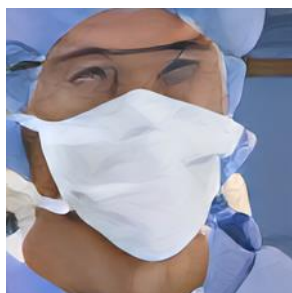
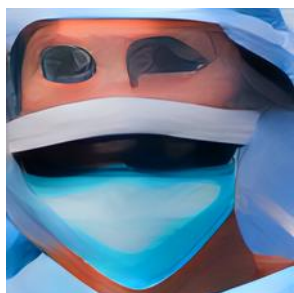
- FMLD 5K: SR และ AFMDK: SR เป็นชุดข้อมูลที่มีการปรับปรุงภาพ
- FMLD 5K: Non_SR และ AFMDK: Non_SR เป็นชุดข้อมูลที่ไม่มีการปรับปรุงภาพ
- FMLD 5K: size < 50 SR และ AFMDK: size < 50 SR เป็นชุดข้อมูลที่มีการปรับปรุงภาพ และมีขนาดเล็กกว่า 50 พิกเซล
- FMLD 5K: size < 50 Non_SR และ AFMDK: size < 50 Non_SR เป็นชุดข้อมูลที่ไม่มีการปรับปรุงภาพและมีขนาดเล็กกว่า 50 พิกเซล
- FMLD 5K: size >= 50 SR และ AFMDK: size >= 50 SR เป็นชุดข้อมูลที่มีการปรับปรุงภาพ และมีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล

- FMLD 5K: size ≥ 50 Non_SR และ AFMDK: size ≥ 50 Non_SR เป็นชุดข้อมูลที่ไม่มีการปรับปรุงภาพและมีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล

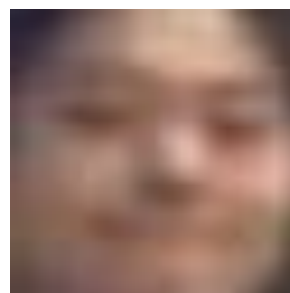
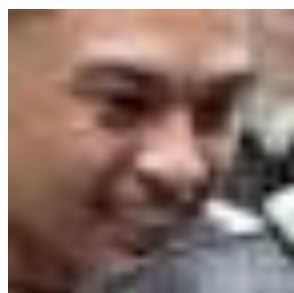
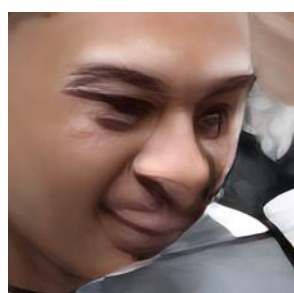
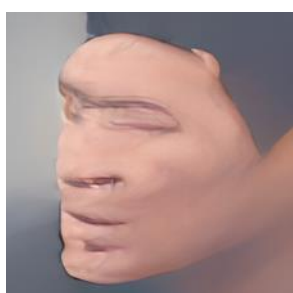


ภาพที่ 65 ขั้นตอนการสร้างชุดข้อมูล ในการทดลองใช้การปรับปรุงภาพ (Image Enhancement) โดยใช้ Super-Resolution (SR)

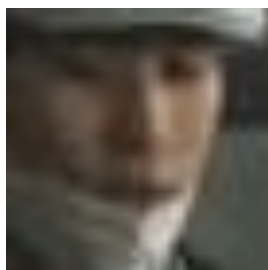
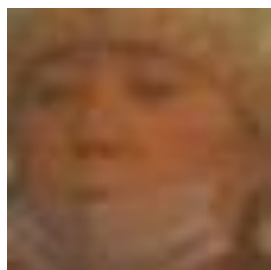
ตัวอย่างผลการปรับปรุงภาพและภาพต้นฉบับภาพขนาดเล็ก (size < 50) ของชุดข้อมูล FMLD 5K ประเภทสวมหน้ากากถูกต้อง ประเภทไม่สวมหน้ากาก และประเภทสวมหน้ากากไม่ถูกต้อง แสดงดังภาพที่ 66 ภาพที่ 67 และภาพที่ 68 ตามลำดับ

(ก) ภาพ
ต้นฉบับ(ข) ผล
การ
ปรับปรุง
ภาพ (SR)

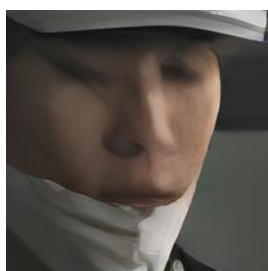
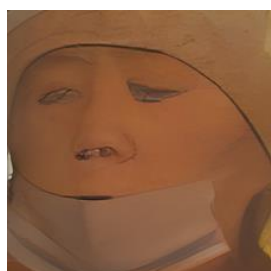
ภาพที่ 66 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากที่ต้องที่มีขนาดเล็กกว่า 50 พิกเซลของชุดข้อมูล FMLD 5K

(ก) ภาพ
ต้นฉบับ(ข) ผล
การ
ปรับปรุง
ภาพ
(SR)

ภาพที่ 67 ตัวอย่างผลการปรับปรุงภาพประเภทไม่สวมหน้ากากที่มีขนาดเล็กกว่า 50 พิกเซลของชุดข้อมูล FMLD 5K



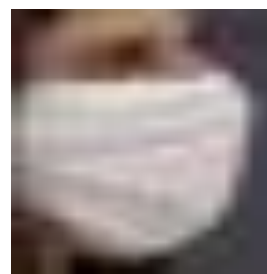
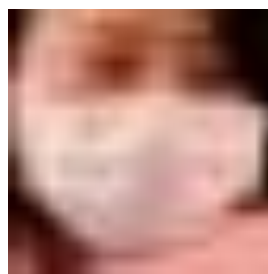
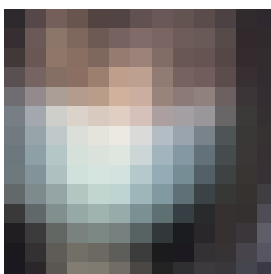
(ก) ภาพ
ต้นฉบับ



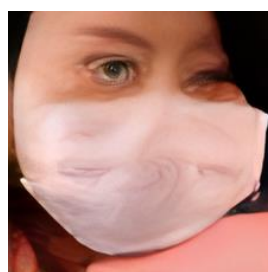
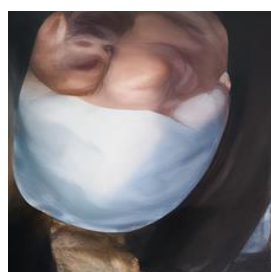
(ข) ผลการ
ปรับปรุง
ภาพ (SR)

ภาพที่ 68 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากไม่ถูกต้องที่มีขนาดเล็กกว่า 50 พิกเซล
ของชุดข้อมูล FMLD 5K

ตัวอย่างผลการปรับปรุงภาพและภาพต้นฉบับภาพขนาดเล็ก (size < 50) ของชุด
ข้อมูล AFMDK ประเภทสวมหน้ากากถูกต้อง ประเภทไม่สวมหน้ากาก และประเภทสวมหน้ากากไม่
ถูกต้องแสดงดัง ภาพที่ 69 ภาพที่ 70 และภาพที่ 71

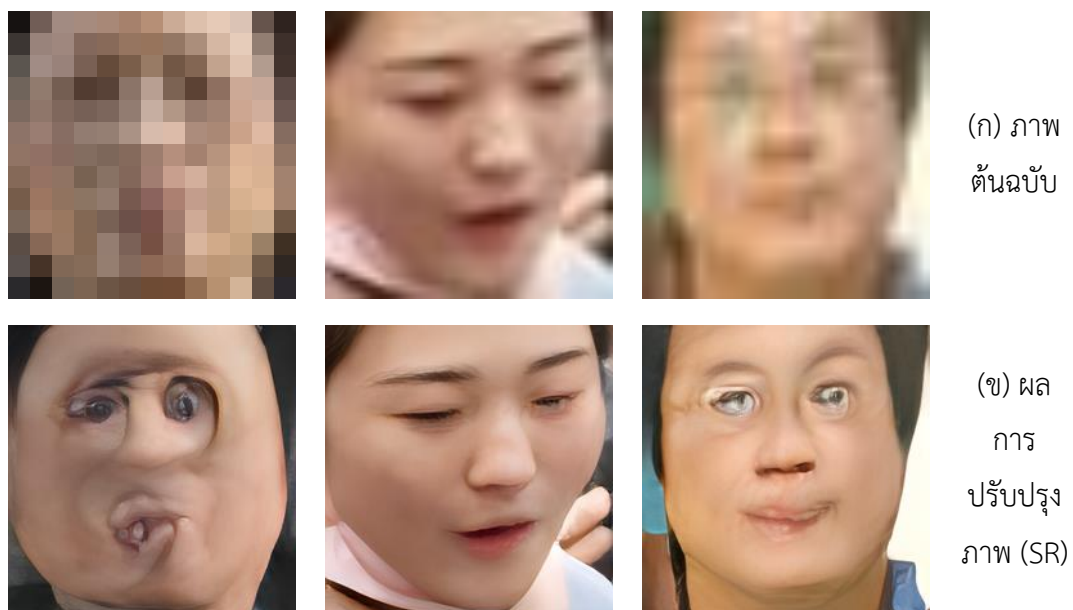


(ก) ภาพ
ต้นฉบับ

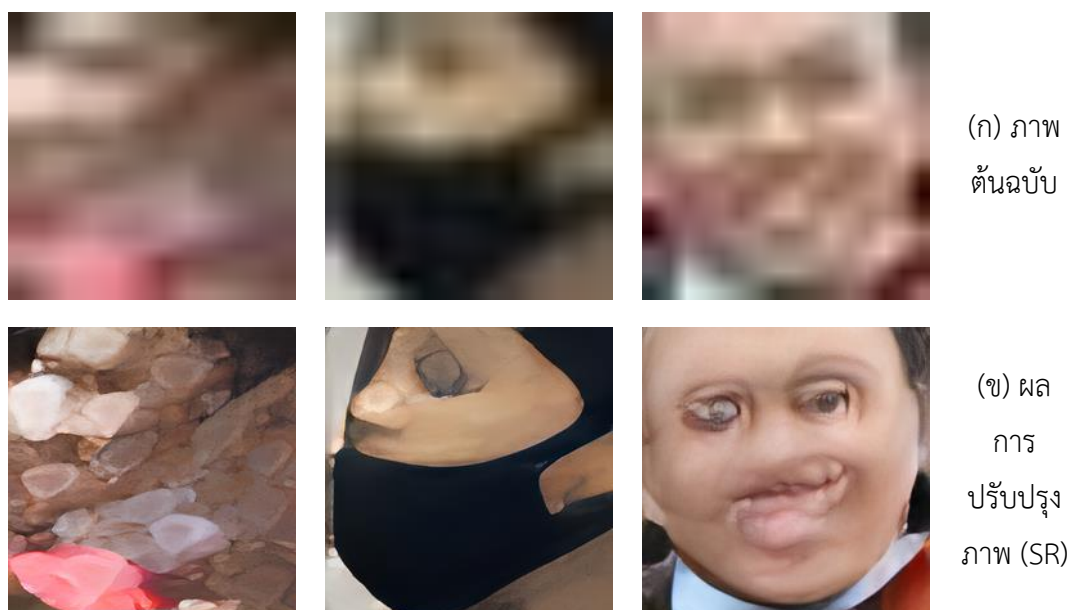


(ข) ผลการ
ปรับปรุง
ภาพ (SR)

ภาพที่ 69 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากถูกต้องที่มีขนาดเล็กกว่า 50 พิกเซลของ
ชุดข้อมูล AFMDK

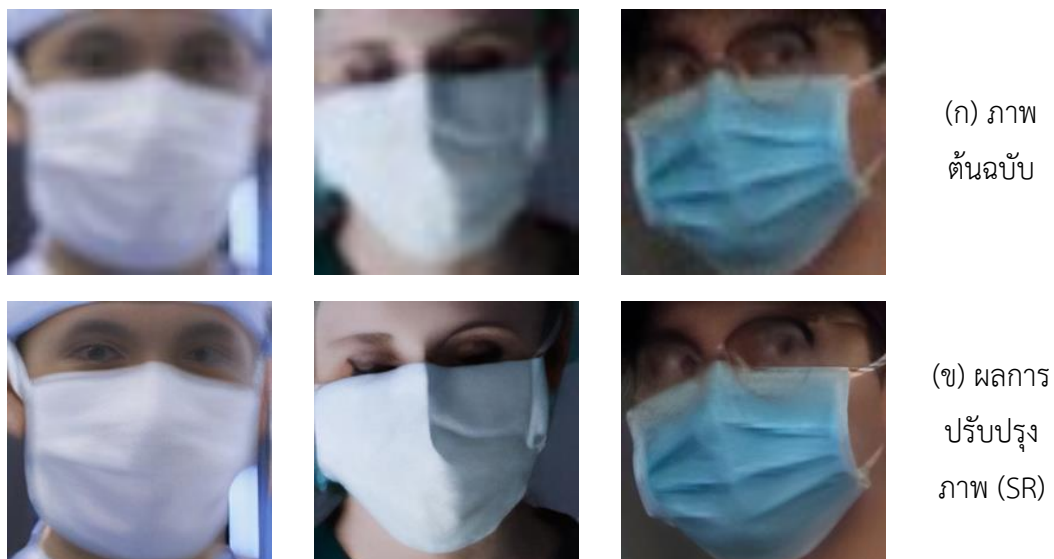


ภาพที่ 70 ตัวอย่างผลการปรับปรุงภาพประเภทไม่สวมหน้ากากที่มีขนาดเล็กกว่า 50 พิกเซลของชุดข้อมูล AFMDK

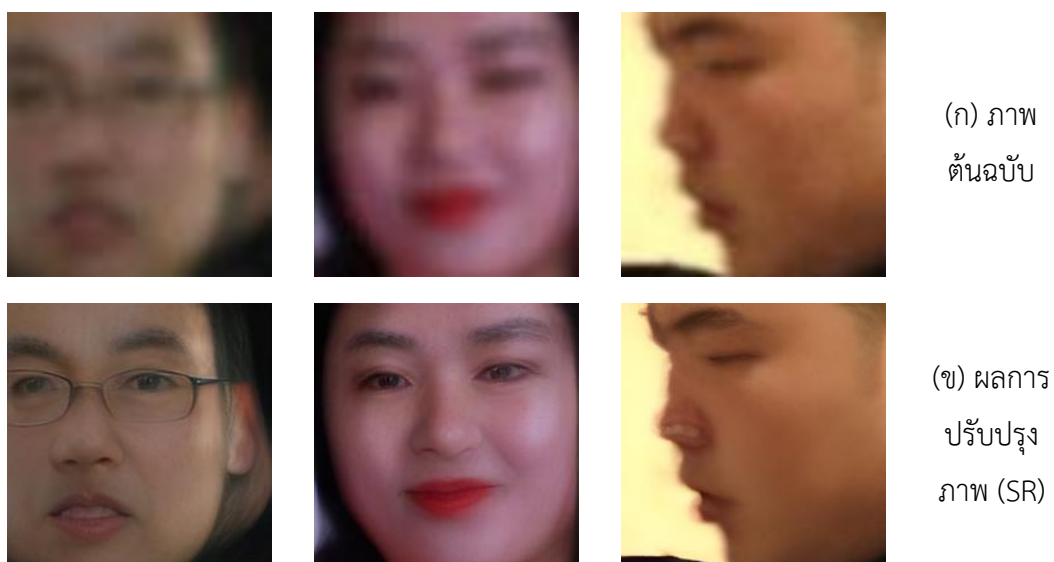


ภาพที่ 71 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากไม่ถูกต้องที่มีขนาดเล็กกว่า 50 พิกเซลของชุดข้อมูล AFMDK

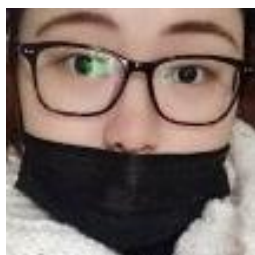
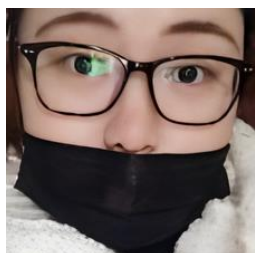
ตัวอย่างผลการปรับปรุงภาพและภาพต้นฉบับภาพขนาดใหญ่ (size ≥ 50) ของชุดข้อมูล FMLD 5K ประเภทสวมหน้ากากถูกต้อง ประเภทไม่สวมหน้ากาก และประเภทสวมหน้ากากไม่ถูกต้อง แสดงดังภาพที่ 72 ภาพที่ 73 และภาพที่ 74 ตามลำดับ



ภาพที่ 72 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากถูกต้องที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซลของชุดข้อมูล FMLD 5K

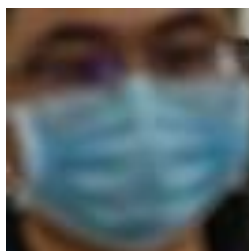
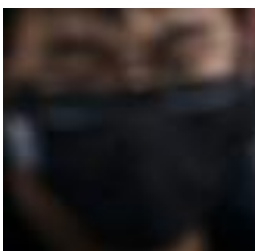
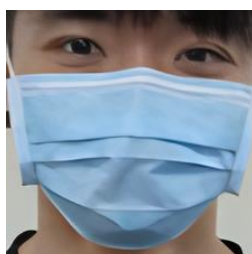
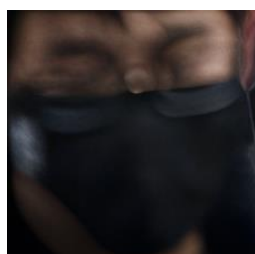


ภาพที่ 73 ตัวอย่างผลการปรับปรุงภาพประเภทไม่สวมหน้ากากที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซลของชุดข้อมูล FMLD 5K

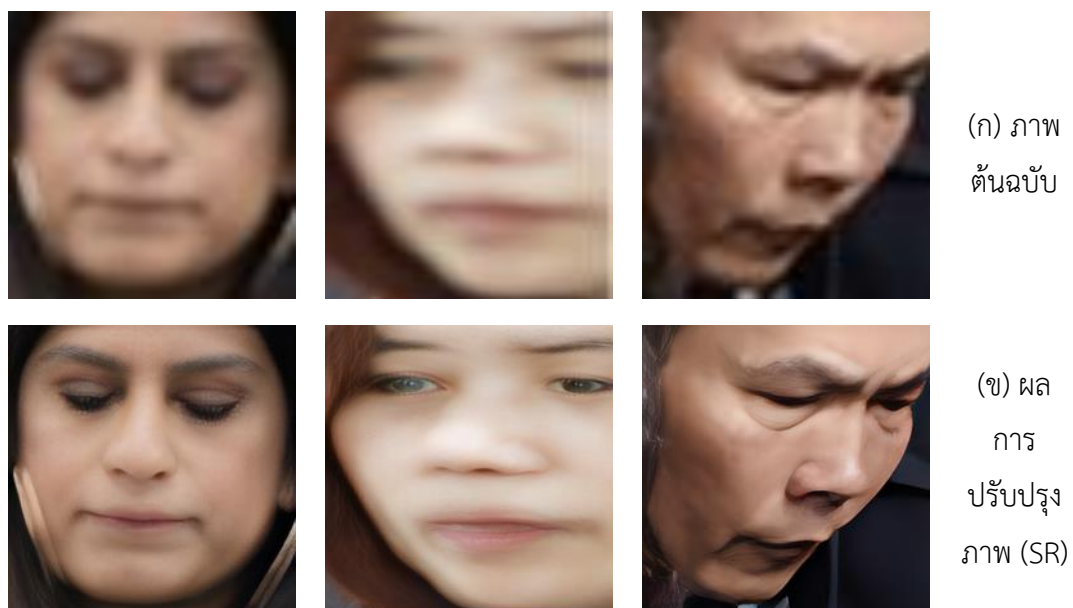
(ก) ภาพ
ต้นฉบับ(ข) ผลการ
ปรับปรุงภาพ
(SR)

ภาพที่ 74 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากไม่ถูกต้องที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซลของชุดข้อมูล FMLD 5K

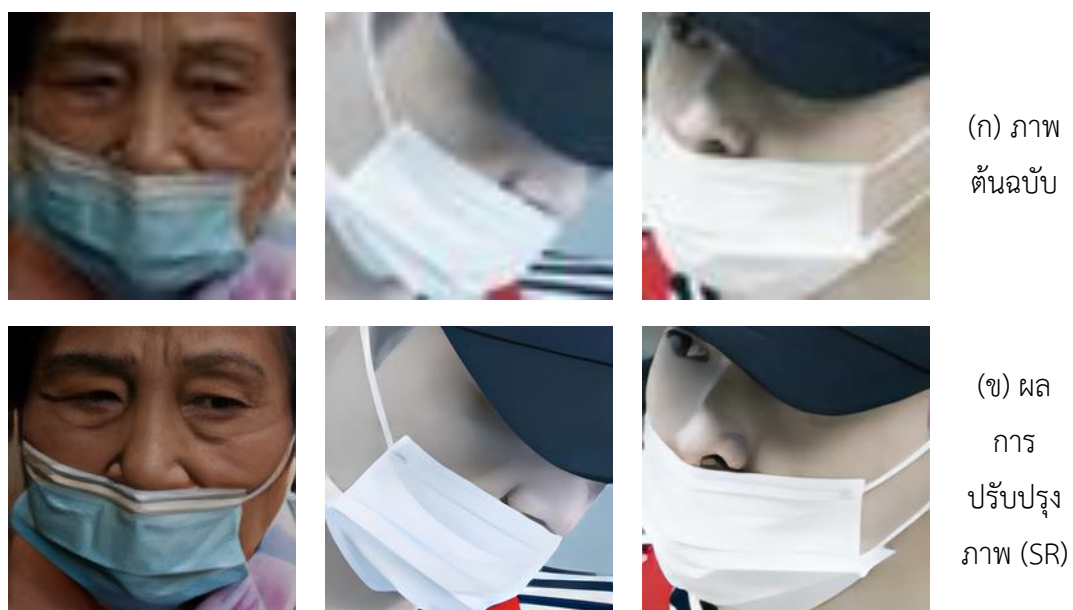
ตัวอย่างผลการปรับปรุงภาพและภาพต้นฉบับภาพขนาดใหญ่ (size ≥ 50) ประเภทสวมหน้ากากถูกต้อง ประเภทไม่สวมหน้ากาก และประเภทสวมหน้ากากไม่ถูกต้องแสดงดังภาพที่ 75 ภาพที่ 76 และภาพที่ 77 จะเห็นว่าภาพขนาดใหญ่ที่ผ่านการปรับปรุงภาพรายละเอียดของภาพจะสูงขึ้นในภาพที่ภาพไม่ชัด

(ก) ภาพ
ต้นฉบับ(ข) ผลการ
ปรับปรุง
ภาพ (SR)

ภาพที่ 75 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากถูกต้องที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซลของชุดข้อมูล AFMDK



ภาพที่ 76 ตัวอย่างผลการปรับปรุงภาพประเภทไม่สวมหน้ากากที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซลของชุดข้อมูล AFMDK



ภาพที่ 77 ตัวอย่างผลการปรับปรุงภาพประเภทสวมหน้ากากไม่ถูกต้องที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซลของชุดข้อมูล AFMDK

4.6.2. เปรียบเทียบประสิทธิภาพของโมเดลเพื่อการปรับปรุงภาพ

ในหัวข้อนี้ทำการศึกษาทดลองและเปรียบเทียบประสิทธิภาพของโมเดลเมื่อมีการปรับปรุงภาพ (SR) กับเมื่อไม่มีการใช้การปรับปรุงภาพ (Non_SR) การประเมินประสิทธิภาพของโมเดลใช้ข้อมูลเข้าเป็นภาพใบหน้าเพียงอย่างเดียวในกรณีข้อมูลเข้าเดี่ยว และภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปากในกรณีข้อมูลเข้าหลายค่า ใช้โมเดลจำแนกเป็น 3 ประเภท คือ ประเภทสวมหน้ากากถูกต้อง ประเภทสวมหน้ากากไม่ถูกต้อง และประเภทไม่สวมหน้ากาก ทำการทดลองบนชุดข้อมูล FMLD 5K และ AFMDK โดยใช้ค่าความถูกต้อง (Accuracy), ค่าความแม่นยำ (Precision), ค่าความครบถ้วน (Recall) และค่า F1 (F1-score) เป็นตัววัดประสิทธิภาพของโมเดล และแสดงผลการทดลองจากชุดข้อมูลทดสอบทั้งหมด ผลการทดลองที่ใช้เพียงภาพในชุดข้อมูลที่มีขนาดเล็กกว่า 50 พิกเซล และผลการทดลองที่ใช้เพียงภาพในชุดข้อมูลที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล

จากผลการทดลองบนชุดข้อมูล FMLD 5K พบว่าการปรับปรุงภาพมีผลทำให้โมเดลข้อมูลเข้าแบบเดี่ยวมีประสิทธิภาพโดยภาพรวมลดลงดังตารางที่ 20 แต่มีผลทำให้โมเดลข้อมูลเข้าหลายค่ามีประสิทธิภาพดีขึ้นแสดงดังตารางที่ 21 จะเห็นได้ว่าในชุดข้อมูล FMLD 5K (3 class) โมเดลข้อมูลเข้าหลายค่ามีความสามารถในการจำแนกเพิ่มขึ้นเมื่อใช้งาน SR ในการปรับปรุงภาพ เมื่อสังเกตประสิทธิภาพของโมเดลแบบแบ่งตามประเภท จะเห็นได้ว่าโมเดลมีประสิทธิภาพดีที่สุดในประเภทสวมหน้ากากถูกต้อง แต่ประสิทธิภาพจะลดลงอย่างเห็นได้ชัดในประเภทไม่สวมหน้ากากและประเภทสวมหน้ากากไม่ถูกต้องแสดงดังตารางที่ 22

ตารางที่ 20 ผลการทดลองโมเดลข้อมูลเข้าเดี่ยว (Single Input) ใช้ชุดข้อมูล FMLD 5K (3 class) โดยใช้โมเดลที่ฝึกโดยภาพ Non_SR และโมเดลที่ฝึกโดยภาพ SR

Dataset	Evaluation Measurement (%)	Non_SR	SR
FMLD 5K (3 class)	Accuracy	84.00	82.73
	Precision	81.99	82.20
	Recall	81.34	79.52
	F1-score	81.12	79.14

ตารางที่ 21 ผลการทดลองโมเดลข้อมูลเข้าหลายค่า (Multiple Input) ใช้ชุดข้อมูล FMLD 5K (3 class) โดยใช้โมเดลที่ฝึกโดยภาพ Non_SR และโมเดลที่ฝึกโดยภาพ SR

Dataset	Evaluation Measurement (%)	Non_SR	SR
FMLD 5K (3 class)	Accuracy	83.27	84.60
	Precision	81.25	82.91
	Recall	80.17	82.10
	F1-score	80.09	81.89

ตารางที่ 22 ผลการทดลองโมเดลข้อมูลเข้าหลายค่า (Multiple Input) ใช้ชุดข้อมูล FMLD 5K (3 class) ของโมเดลที่ฝึกโดยภาพ SR แบบแบ่งตามประเภท

Dataset	Class	Evaluation Measurement (%)		
		precision	recall	f1-score
FMLD 5K (3 class)	mask_worned_incorrect	68.27	82.26	74.62
	with_mask	90.70	93.60	92.13
	without_mask	89.76	70.45	78.94
	Average	82.91	82.10	81.89

ในการทดลองบนชุดข้อมูล AFMDK พบว่าการปรับปรุงภาพส่งผลให้ประสิทธิภาพของทั้งโมเดลข้อมูลเข้าเดี่ยวและโมเดลข้อมูลเข้าหลายค่ามีประสิทธิภาพลดลงดังตารางที่ 23 และตารางที่ 24 เนื่องจากในชุดข้อมูล AFMDK มีสัดส่วนของจำนวนภาพขนาดเล็กค่อนข้างสูง เมื่อสังเกตประสิทธิภาพของโมเดลแบบแบ่งตามประเภท จะเห็นได้ว่าโมเดลมีประสิทธิภาพดีในประเภทสวมหน้ากากถูกต้องและประเภทไม่สวมหน้ากาก แต่โมเดลไม่สามารถจำแนกประเภทสวมหน้ากากไม่ถูกต้องได้อย่างถูกต้องแสดงดังตารางที่ 25

ตารางที่ 23 ผลการทดลองโมเดลข้อมูลเข้าเดี่ยว (Single Input) ใช้ชุดข้อมูล AFMDK (3 class) โดยใช้โมเดลที่ฝึกโดยภาพ Non_SR และโมเดลที่ฝึกโดยภาพ SR

Dataset	Evaluation Measurement (%)	Non_SR	SR
AFMDK (3 class)	Accuracy	96.40	92.23
	Precision	91.24	73.56
	Recall	83.03	81.17
	F1-score	86.01	76.86

ตารางที่ 24 ผลการทดลองโมเดลข้อมูลเข้าหลายค่า (Multiple Input) ใช้ชุดข้อมูล AFMDK (3 class) โดยใช้โมเดลที่ฝึกโดยภาพ Non_SR และโมเดลที่ฝึกโดยภาพ SR

Dataset	Evaluation Measurement (%)	Non_SR	SR
AFMDK (3 class)	Accuracy	96.40	94.44
	Precision	91.02	79.15
	Recall	82.08	72.68
	F1-score	85.23	74.81

ตารางที่ 25 ผลการทดลองโมเดลข้อมูลเข้าหลายค่า (Multiple Input) ใช้ชุดข้อมูล AFMDK (3 class) ของโมเดลที่ฝึกโดยภาพ SR แบบแบ่งตามประเภท

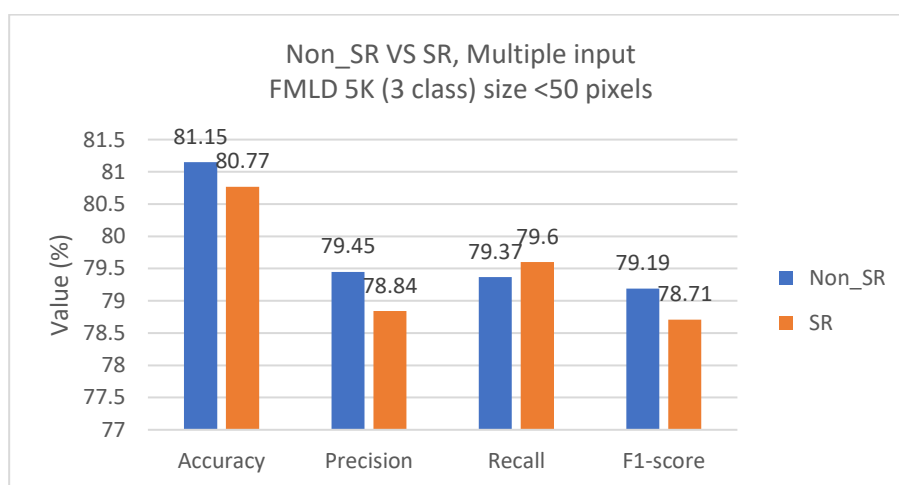
Dataset	Class	Evaluation Measurement (%)		
		precision	recall	f1-score
AFMDK (3 class)	mask_wearred_incorrect	52.63	29.41	37.74
	with_mask	96.57	97.35	96.96
	without_mask	88.26	91.26	89.74
	Average	79.15	72.68	74.81

ต่อไปนี้จะเป็นการพิจารณาประสิทธิภาพกรณีทดสอบโมเดลข้อมูลเข้าหลายค่า เฉพาะภาพที่มีขนาดเล็ก จากผลการทดลองบนชุดข้อมูล FMLD 5K พบว่าผลลัพธ์ที่ได้ คือ SR ทำให้ประสิทธิภาพลดลงแสดงดังตารางที่ 26 และกราฟในภาพที่ 78 สำหรับผลการทดลองบนชุดข้อมูล

AFMDK พบว่า SR ทำให้ประสิทธิภาพลดลงเช่นกัน ดังแสดงในตารางที่ 27 และกราฟในภาพที่ 79 ทั้งนี้เนื่องจากการใช้เทคนิค SR ในการปรับปรุงภาพที่มีขนาดเล็ก ทำให้เกิดการบิดเบี้ยวของภาพได้

ตารางที่ 26 ผลการเปรียบเทียบโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล FMLD 5K (3 class) ขนาดเล็กกว่า 50 พิกเซล ที่ฝึกและทดสอบโดยใช้ภาพ Non_SR และ SR

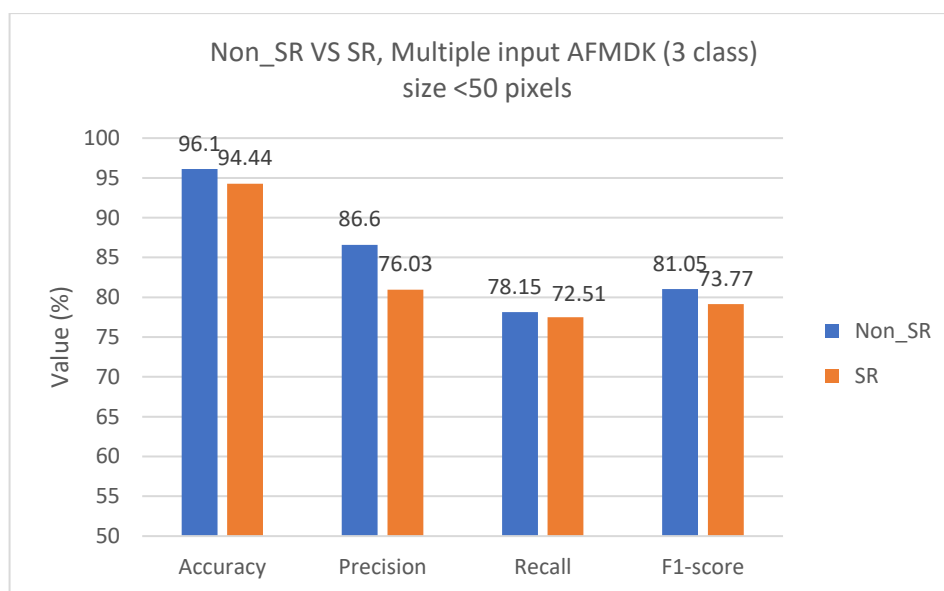
Dataset	Evaluation Measurement (%)	Train by Non_SR (Multiple Input)	Train by SR (Multiple Input)
		Non_SR	SR
FMLD 5K (3 class) size <50 pixels	Accuracy	81.15	80.77
	Precision	79.45	78.84
	Recall	79.37	79.60
	F1-score	79.19	78.71



ภาพที่ 78 กราฟประสิทธิภาพ Non_SR และ SR ของโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล FMLD 5K (3 class) ทดสอบโดยภาพขนาดเล็กกว่า 50 พิกเซล

ตารางที่ 27 ผลการเปรียบเทียบโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล AFMDK (3 class) ที่คัดเลือกภาพที่มีขนาดเล็กกว่า 50 พิกเซล ที่ฝึกและทดสอบโดยใช้ภาพ Non_SR และ SR

Dataset	Evaluation Measurement (%)	Train by Non_SR (Multiple Input)	Train by SR (Multiple Input)
		Non_SR	SR
AFMDK (3 class) size <50 pixels	Accuracy	96.10	94.44
	Precision	86.60	76.03
	Recall	78.15	72.51
	F1-score	81.05	73.77

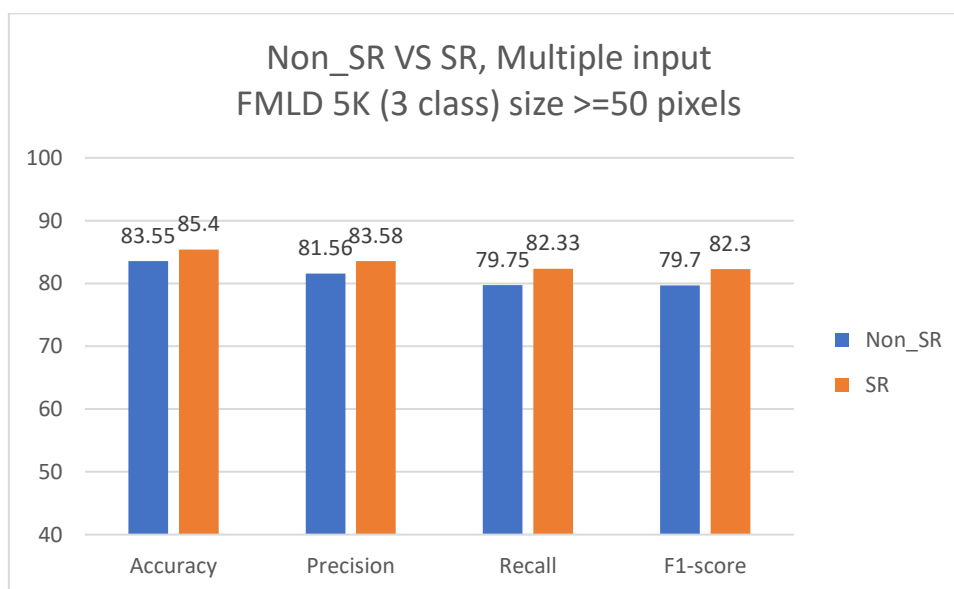


ภาพที่ 79 กราฟประสิทธิภาพ Non_SR และ SR ของโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล AFMDK (3 class) ทดสอบโดยภาพขนาดเล็กกว่า 50 พิกเซล

ต่อไปเป็นการพิจารณาประสิทธิภาพกรณีทดสอบโมเดลข้อมูลเข้าหลายค่าเฉพาะภาพที่มีขนาดมากกว่าหรือเท่ากับ 50 พิกเซล จากผลการทดลองบนชุดข้อมูล FMLD 5K พบว่า SR ทำให้ประสิทธิภาพดีขึ้นแสดงดังตารางที่ 28 และกราฟในภาพที่ 80 สำหรับผลการทดลองบนชุดข้อมูล AFMDK พบว่า SR ทำให้ประสิทธิภาพดีขึ้นเช่นกัน ดังตารางที่ 29 และกราฟในภาพที่ 81

ตารางที่ 28 ผลการเปรียบเทียบโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล FMLD 5K (3 class) ขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล 50 พิกเซล ที่ฝึกและทดสอบโดยใช้ภาพ Non_SR และ SR

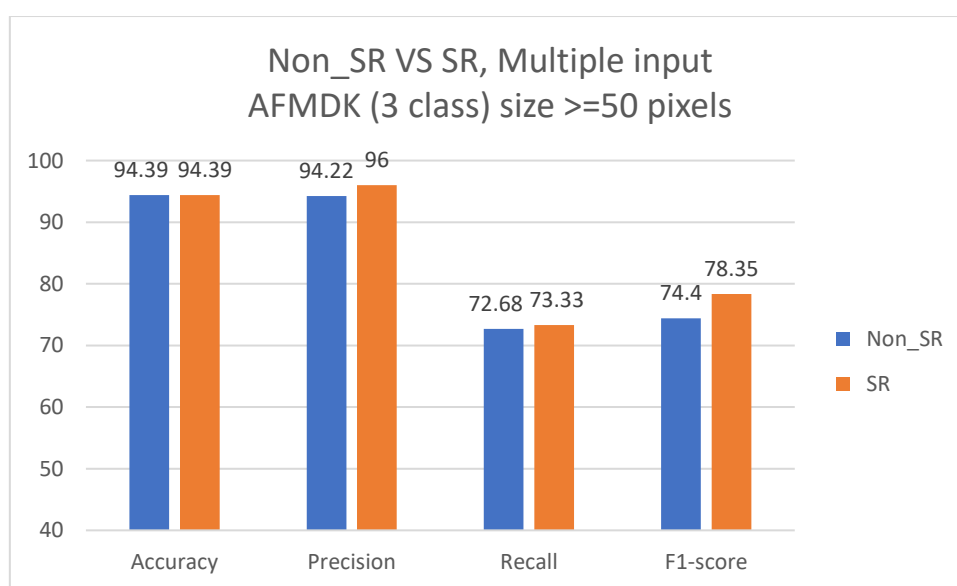
Dataset	Evaluation Measurement (%)	Train by Non_SR (Multiple Input)	Train by SR (Multiple Input)
		Non_SR	SR
FMLD 5K (3 class) size >=50 pixels	Accuracy	83.55	85.40
	Precision	81.56	83.58
	Recall	79.75	82.33
	F1-score	79.70	82.30



ภาพที่ 80 กราฟประสิทธิภาพ Non_SR และ SR ของโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล FMLD 5K (3 class) ทดสอบโดยภาพขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล

ตารางที่ 29 ผลการเปรียบเทียบโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล AFMDK (3 class) ที่คัดเลือกภาพที่มีขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล ที่ฝึกและทดสอบโดยใช้ภาพ Non_SR และ SR

Dataset	Evaluation Measurement (%)	Train by Non_SR (Multiple Input)	Train by SR (Multiple Input)
		Non_SR	SR
AFMDK (3 class) size ≥ 50 pixels	Accuracy	94.39	94.39
	Precision	94.22	96.00
	Recall	72.68	73.33
	F1-score	74.40	78.35



ภาพที่ 81 กราฟประสิทธิภาพ Non_SR และ SR ของโมเดลข้อมูลเข้าหลายค่าที่ใช้ชุดข้อมูล AFMDK (3 class) ทดสอบโดยภาพขนาดใหญ่กว่าหรือเทียบเท่า 50 พิกเซล

จากผลการทดลองเพื่อประเมินประสิทธิภาพของโมเดลข้อมูลเข้าหลายค่าเมื่อมีการใช้การปรับปรุงภาพที่ได้กล่าวมาข้างต้น จะได้ว่า ถ้าภาพมีขนาดไม่เล็กเกินไป กล่าวคือขนาดมากกว่าหรือเท่ากับ 50 พิกเซลแล้วการปรับปรุงภาพ SR ช่วยให้ประสิทธิภาพดีขึ้น แต่ถ้าภาพมีขนาดเล็กกว่าคือเล็กกว่า 50 พิกเซลแล้วการปรับปรุงภาพ SR จะทำให้ประสิทธิภาพลดลง เมื่อพิจารณาประสิทธิภาพโดยรวมทั้งชุดข้อมูล FMLD 5K และ AFMDK พบว่าการปรับปรุงภาพ SR ช่วยเพิ่มประสิทธิภาพบนชุดข้อมูล FMLD 5K แต่กลับลดประสิทธิภาพลงในชุดข้อมูล AFMDK

สาเหตุที่ชุดข้อมูล FMLD 5K และชุดข้อมูล AFMDK มีประสิทธิภาพต่างกันเป็นเพราะจำนวนภาพที่มีขนาดเล็กกว่า 50 พิกเซล และภาพที่มีขนาดเล็กกว่า 30 พิกเซลแสดงดังตารางที่ 18 และตารางที่ 19 ชุดข้อมูล FMLD 5K จะมีภาพขนาดเล็กจำนวนไม่มาก เมื่อเทียบกับภาพทั้งหมด ในขณะที่ชุดข้อมูล AFMDK จะมีภาพขนาดเล็กเป็นจำนวนมาก และหลายภาพมีขนาดเล็กกว่า 30 พิกเซล

โมเดลข้อมูลเข้าหลายค่า (Multiple Input) สามารถช่วยเพิ่มประสิทธิภาพในการจำแนกประเภทภาพใบหน้าสวมหน้ากาก การปรับปรุงภาพโดยใช้โมเดล BSRGANs ให้ภาพมีรายละเอียดมากขึ้น ไม่สามารถช่วยเพิ่มประสิทธิภาพโมเดลต่อภาพขนาดเล็ก เมื่อภาพมีขนาดเล็กเกินไปจะส่งผลให้ภาพมีการเปลี่ยนแปลง และส่งผลต่อประสิทธิภาพของโมเดลโดยตรง แต่สามารถเพิ่มประสิทธิภาพของโมเดลต่อภาพขนาดใหญ่ได้

สำหรับการวัดความเร็วการทำงานของโมเดล จากการสุ่มภาพมา 6 ภาพที่มีขนาดของภาพแตกต่างกันมาวัดความเร็วโดยใช้โมเดลข้อมูลเข้าหลายค่าที่มีข้อมูลเข้าเป็นภาพใบหน้า จมูก และปาก โดยโมเดลตาม Algorithm1 (ไม่มีการปรับปรุงภาพ) ในการประมวลผลภาพหนึ่งภาพจะมีความเร็วเฉลี่ยอยู่ที่ 0.83 วินาที และโมเดลตาม Algorithm2 (มีการปรับปรุงภาพ) จะมีความเร็วเฉลี่ยอยู่ที่ 1.26 วินาที ความเร็วในการทำงานของโมเดลตาม Algorithm1 และความเร็วในการทำงานของโมเดลตาม Algorithm2 แสดงดังตารางที่ 30

ตารางที่ 30 แสดงเวลาที่ใช้ในการทำงานของโมเดลตาม Algorithm1 (ไม่ปรับปรุงภาพ) และโมเดลตาม Algorithm2 (ปรับปรุงภาพ)

ภาพ	ขนาดภาพต้นฉบับ	ความเร็วการทำงาน (s)	
		โมเดล Algorithm1	โมเดล Algorithm2
ภาพที่ 1	600x480	0.73	1.12
ภาพที่ 2	455x513	0.76	0.89
ภาพที่ 3	1536x1024	1.00	1.75
ภาพที่ 4	1536x1024	1.06	2.02
ภาพที่ 5	180x180	0.59	0.68
ภาพที่ 6	517x690	0.82	1.11
-	Average	0.83	1.26

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1. สรุปผลการวิจัย

งานวิจัยนี้นำเสนอขั้นตอนวิธีการใช้โมเดลข้อมูลเข้าหลายค่าเพื่อเพิ่มประสิทธิภาพของโมเดลในการจำแนกใบหน้าสวมหน้ากาก โดยการสังเกตข้อแตกต่างของภาพในแต่ละประเภท คือ ภาพประเภทสวมหน้ากากถูกต้องจะมีหน้ากากปิดทั้งบริเวณจมูกและปาก ในขณะที่ภาพประเภทไม่ใส่หน้ากากจะเห็นทั้งบริเวณจมูกและปากในกรณีทั่วไป และภาพประเภทใส่หน้ากากไม่ถูกต้องจะเห็นบริเวณจมูกหรือปากอย่างใดอย่างหนึ่ง หรือทั้งคู่ ใช้ชุดข้อมูล 2 ชุด คือ ชุดข้อมูล FMLD และชุดข้อมูล AFMDK โดยดำเนินการหาภาพใบหน้า ภาพบริเวณจมูกและภาพบริเวณปากเพื่อใช้เป็นข้อมูลเข้าให้กับโมเดล โดยใช้โมเดล MTCNN ทำให้ได้ชุดข้อมูลที่มีทั้งภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก ผลการทดลองโมเดลข้อมูลเข้าหลายค่ามีประสิทธิภาพสูงกว่าโมเดลข้อมูลเดี่ยวทั้งค่าความถูกต้อง ค่าความแม่นยำ ค่าความครบถ้วน และค่า F1 การนำเสนอการใช้เทคนิค Super-Resolution เพื่อใช้ในการปรับปรุงภาพในกรณีที่ภาพมีความคมชัดต่ำหรือมีขนาดเล็ก จากผลการทดลองการใช้งานโมเดล BSRGANs ไม่สามารถช่วยแก้ปัญหาภาพขนาดเล็กได้ แต่สามารถเพิ่มประสิทธิภาพของโมเดลต่อภาพขนาดใหญ่ได้

5.2. ปัญหาและอุปสรรค

ขั้นตอนวิธีที่เสนอใช้โมเดล MTCNN เพื่อตรวจจับภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปาก ทำให้ประสิทธิภาพของโมเดลขึ้นอยู่กับภาพที่ได้รับมาจากโมเดลที่ใช้ตรวจจับภาพใบหน้า ภาพบริเวณจมูก และภาพบริเวณปากโดยตรง อีกทั้งการใช้โมเดลข้อมูลเข้าหลายค่าที่ใช้ภาพเพียงอย่างเดียวจะใช้ทรัพยากรและเวลาในการฝึกมากขึ้น เนื่องจากภาพทั้ง 3 ภาพจะถูกนำไปฝึกเพื่อสกัดคุณลักษณะตามประเภทของภาพ

5.3. ข้อเสนอแนะ

สามารถนำแนวคิดนี้ไปพัฒนาต่อโดยไม่จำเป็นต้องหาภาพบริเวณจมูก และภาพบริเวณปาก แต่ใช้โมเดลอื่นเพื่อหาว่าเห็นบริเวณจมูก และปากหรือไม่ แล้วส่งข้อมูลเป็นชุดตัวเลขแทน เพื่อลดทรัพยากรและเวลาในการฝึก หรือนำไปประยุกต์ใช้กับงานอื่น เช่น การตรวจจับการสวมแว่นนिरภัยในงานที่ต้องมีการใส่ว่านนिरภัยโดยใช้ข้อมูลเข้าเป็นภาพใบหน้า และภาพบริเวณตา เป็นต้น

เอกสารอ้างอิง

- [1] J. Brownlee, "{A Gentle Introduction to Computer Vision}," 2019. [Online]. [Accessed 8 10 2021].
- [2] S. Hayat, S. Kun, Z. Tengtao, Y. Yu, T. Tu and Y. Du, "A deep learning framework using convolutional neural network for multi-class object recognition," in *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, 2018.
- [3] J. Brownlee, "{A Gentle Introduction to Object Recognition With Deep Learning}," 2021. [Online]. Available: <https://machinelearningmastery.com/object-recognition-with-deep-learning/>. [Accessed 4 8 2021].
- [4] J. Walsh, N. O' Mahony, S. Campbell, A. Carvalho, L. Krpalkova, G. Velasco-Hernandez, S. Harapanahalli and D. Riordan, "Deep Learning vs. Traditional Computer Vision," 2019.
- [5] Z.-Q. Zhao, P. Zheng, S.-T. Xu and X. Wu, "Object Detection With Deep Learning: A Review," vol. 30, pp. 3212-3232, 2019.
- [6] M. R. Bhuiyan, S. A. Khushbu and M. S. Islam, "A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, India, 2020.
- [7] J. Gathani and K. Shah, "Detecting Masked Faces using Region-based Convolutional Neural Network," in *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, 2020.
- [8] S. Srinivasan, R. Rujula Singh, R. R. Biradar and S. Revathi, "COVID-19 Monitoring System using Social Distancing and Face Mask Detection on Surveillance video datasets," in *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2021.
- [9] TensorFlow, "The Functional API," 2021. [Online]. Available: https://www.tensorflow.org/guide/keras/functional#models_with_multiple_inputs_and_outputs. [Accessed 1 5 2022].

- [10] J. Zhang, F. Han, Y. Chun and W. Chen, "A Novel Detection Framework About Conditions of Wearing Face Mask for Helping Control the Spread of COVID-19," *IEEE Access*, vol. 9, pp. 42975 - 42984, 2021.
- [11] LARXEL, "Face Mask Detection," Kaggle, [Online]. Available: <https://www.kaggle.com/datasets/andrewmvd/face-mask-detection>.
- [12] C. Tang, Y. Ling, X. Yang and W. Jin, "Multi-View Object Detection Based on Deep Learning," *Applied Sciences*, vol. 8, p. 1423, 8 2018.
- [13] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- [14] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [15] X. Wang, Z. Chen, B. Wei and M. Ling, "Application of Pruning Yolo-V4 with Center Loss in Mask Wearing Recognition for Gymnasiums and Sports Grounds of Colleges and Universities," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020.
- [16] S. Cristina, "{Calculus in Action: Neural Networks}," 2021. [Online]. Available: <https://machinelearningmastery.com/calculus-in-action-neural-networks/>. [Accessed 9 10 2021].
- [17] S. Asiri, "{Machine Learning Classifiers}," 2018. [Online]. Available: <https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623>. [Accessed 9 10 2021].
- [18] S. Saha, "{A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way}," 2018. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. [Accessed 9 10 2021].
- [19] J. Brownlee, "{What is Deep Learning?}," 2020. [Online]. Available: <https://machinelearningmastery.com/what-is-deep-learning/>. [Accessed 9 10 2021].

- [20] A. Anwar, "{Difference between AlexNet, VGGNet, ResNet, and Inception}," 2019. [Online]. Available: <https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaaecccc96>. [Accessed 28 8 2021].
- [21] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [22] J. Brownlee, "{How to Manually Scale Image Pixel Data for Deep Learning}," 2019. [Online]. Available: <https://machinelearningmastery.com/how-to-manually-scale-image-pixel-data-for-deep-learning/>. [Accessed 9 10 2021].
- [23] W. Kim, "Low-Light Image Enhancement: A Comparative Review and Prospects," in *IEEE Access*, 2022.
- [24] S. Yang, Y. Tian, M. Zheng, Y. Du, H. Chen, F. Song, X. Gao and L. Li, "A Review of Image Enhancement Technology Research," in *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, 2021.
- [25] K. Singh, A. Seth, H. S. Sandhu and K. Samdani, "A Comprehensive Review of Convolutional Neural Network based Image Enhancement Techniques," in *2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2019.
- [26] S. Zhou, K. C. K. Chan, C. Li and C. C. Loy, "Towards Robust Blind Face Restoration with Codebook Lookup Transformer," in *Advances in Neural Information Processing Systems*, 2022.
- [27] K. Zhang, J. Liang, L. V. Gool and R. Timofte, "Designing a Practical Degradation Model for Deep Blind Image Super-Resolution," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4771-4780, 2021.
- [28] P. P. Ippolito, "{Feature Extraction Techniques}," 2019. [Online]. Available: <https://towardsdatascience.com/feature-extraction-techniques-d619b56e31be>. [Accessed 9 10 2021].

- [29] H. Brar, "{Object Localization}," 2018. [Online]. Available: <https://medium.com/@harman4422/object-localization-bd314d7e648f>. [Accessed 9 10 2021].
- [30] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001.
- [31] shekharpandey, "Dlib 68 points Face landmark Detection with OpenCV and Python,"study tonight, 22 5 2023. [Online]. Available: <https://www.studytonight.com/post/dlib-68-points-face-landmark-detection-with-opencv-and-python>. [Accessed 18 9 2023].
- [32] D. L. Han Sheng, "Haar Cascades Classifier — A Light Weight Face Detection Technique,"Medium,182021.[Online].Available: <https://hansheng0512.medium.com/haar-cascades-classifier-a-light-weight-face-detection-technique-931b65537a99>. [Accessed 18 9 2023].
- [33] R. Patel, P. Patel, A. Acharya, J. Naik and J. Thaker, "Exploring Facial Landmark Detection Techniques for Attention Detection in Human-Computer Interactions," in *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, 2013.
- [34] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.
- [35] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An Enhanced MobileNet Architecture," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2019.
- [36] J. Brownlee, "{A Gentle Introduction to Transfer Learning for Deep Learning},"2021.[Online].Available:<https://machinelearningmastery.com/transfer-learning-for-deep-learning/>. [Accessed 6 8 2021].

- [37] J. Brownlee, "{Train-Test Split for Evaluating Machine Learning Algorithms}," 2020. [Online]. Available: <https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/>. [Accessed 9 10 2021].
- [38] R. J. Tan, "{Breaking Down Mean Average Precision (mAP)}," 2019. [Online]. Available: <https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52#1a59>. [Accessed 5 9 2021].
- [39] S. Yohanandan, "{mAP (mean Average Precision) might confuse you!}," 2020. [Online]. Available: <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>. [Accessed 5 9 2021].
- [40] I. Buciu, "Color quotient based mask detection," in *2020 International Symposium on Electronics and Telecommunications (ISETC)*, 2020.
- [41] R. K. Kodali and R. Dhanekula, "Face Mask Detection Using Deep Learning," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 2021.
- [42] B. Batagelj, P. Peer, V. Štruc and S. Dobrišek, "How to Correctly Detect Face-Masks for COVID-19 from Visual Information?," *Applied Sciences*, vol. 11, p. 2070, 2 2021.
- [43] S. Abbasi, H. Abdi and A. Ahmadi, "A Face-Mask Detection Approach based on YOLO Applied for a New Collected Dataset," in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, 2021.
- [44] S. Sakshi, A. K. Gupta, S. Singh Yada and U. Kumar, "Face Mask Detection System using CNN," in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2021.
- [45] J. R. V. Jeny, B. Shraddha, B. Ashritha, D. S. Sai and M. Naveen, "Deep learning framework for face mask detection," in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2021.

- [46] J. Lin, "Accurate and Fast Mask Recognition Based on Multiple Color Areas Detection and Face Landmarks Locating," in *2022 IEEE 22nd International Conference on Communication Technology (ICCT)*, 2022.
- [47] L. Shuangyan and G. Huayong, "Lighter and Faster Face Mask Detection Method Based on YOLOv5," in *2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2023.
- [48] W. Vijitkunsawat and P. Chantngarm, "Study of the Performance of Machine Learning Algorithms for Face Mask Detection," in *2020 - 5th International Conference on Information Technology (InCIT)*, 2020.
- [49] W. Han, Z. Huang, A. kuerban, M. Yan and H. Fu, "A Mask Detection Method for Shoppers Under the Threat of COVID-19 Coronavirus," in *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, 2020.
- [50] N. Youssry and A. Khattab, "Accurate Real-Time Face Mask Detection Framework Using YOLOv5," in *2022 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS)*, 2022.
- [51] G. Yang , W. Feng, J. Jin, Q. Lei, X. Li, G. Gui and W. Wang, "Face Mask Recognition System with YOLOV5 Based on Image Recognition," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020.
- [52] S. A. Sanjaya and S. Adi Rakhmawan, "Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, 2020.
- [53] M. R. Karim Sujon, M. R. Hossain, M. J. Al Amin, C. Bepery and M. M. Rahman, "Real-time face mask detection for COVID-19 prevention," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 2022.

- [54] I. B. Venkateswarlu, J. Kakarla and S. Prakash, "Face mask detection using MobileNet and Global Pooling Block," in *2020 IEEE 4th Conference on Information & Communication Technology (CICT)*, 2020.
- [55] A. Oumina, N. El Makhf and M. Hamdi, "Control The COVID-19 Pandemic: Face Mask Detection Using Transfer Learning," in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, 2020.
- [56] P. Reddy, M. Nandini, E. Mamatha, K. Reddy and A. Vishant, "Face mask detection using machine learning techniques," in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2021.
- [57] R. Liu and Z. Ren, "Application of Yolo on Mask Detection Task," in *2021 IEEE 13th International Conference on Computer Research and Development (ICCRD)*, 2021.
- [58] K. Podbucki, J. Suder, T. Marciniak and A. Dąbrowski, "CCTV based system for detection of anti-virus masks," in *2020 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, 2020.
- [59] J. leamsaard, S. N. Charoensook and S. Yammen, "Deep Learning-based Face Mask Detection Using YoloV5," in *2021 9th International Electrical Engineering Congress (iEECON)*, 2021.
- [60] S. Yohanandan, "{mAP (mean Average Precision) might confuse you!}," 2020. [Online]. Available: <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>. [Accessed 5 9 2021].
- [61] S. Ge, J. Li, Q. Ye and Z. Luo, "Detecting masked faces in the wild with lle-cnns," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [62] S. Yang, P. Luo, C. C. Loy and X. Tang, "WIDER FACE: A Face Detection Benchmark," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

- [63] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. Lecun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," *{International Conference on Learning Representations (ICLR) (Banff)}*, 12 2013.
- [64] A. A. Almisreb, N. Jamil and N. M. Din, "Utilizing AlexNet Deep Transfer Learning for Ear Recognition," in *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, 2018.
- [65] L. Aziz, M. S. B. Haji Salam, U. U. Sheikh and S. Ayub, "Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review," *IEEE Access*, vol. 8, pp. 170461-170495, 2020.

ภาคผนวก ก

ผลงานตีพิมพ์เผยแพร่จากวิทยานิพนธ์

P. Rujirapong, W. Wettayaprasit, N. Wattanakitrungroj, “The Multiple Inputs Ensembling for Face Mask Classification Using CNN,” in The Fourteen National Conference on Information Technology (NCIT2022), 2020, pp.1-6

การผสมรวมข้อมูลเข้าหลายค่าเพื่อจำแนกใบหน้าสวมหน้ากากโดยใช้ CNN

The Multiple Inputs Ensembling for Face Mask Classification Using CNN

ภิรมกานต์ รุจิราพงศ์
Peemakarn Rujirapong
สาขาวิทยาศาสตร์การคำนวณ
คณะวิทยาศาสตร์
มหาวิทยาลัยสงขลานครินทร์
Computational Science Division
Faculty of Science
Prince of Songkhla University
Songkhla, Thailand
6310220004@psu.ac.th

วิภาดา เวทย์ประสิทธิ์
Wiphada Weltayaprasit
สาขาวิทยาศาสตร์การคำนวณ
คณะวิทยาศาสตร์
มหาวิทยาลัยสงขลานครินทร์
Computational Science Division
Faculty of Science
Prince of Songkhla University
Songkhla, Thailand
wiphada.w@psu.ac.th

นิวรรณ วัฒนกิจรุ่งโรจน์
Niwan Wattanakitrunroj
คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
Faculty of Information Technology
King Mongkut's University of
Technology Thonburi
Thonburi
Thailand
niwan.watt@kmutt.ac.th

บทคัดย่อ — การตรวจจับวัตถุสามารถนำมาประยุกต์ใช้งานได้หลากหลายรูปแบบ การจำแนกใบหน้าว่าใบหน้านั้นใส่หน้ากากถูกต้อง ใส่หน้ากากไม่ถูกต้องหรือไม่ใส่หน้ากากเป็นการประยุกต์ใช้งานที่เป็นประโยชน์มากในสถานการณ์ปัจจุบัน งานวิจัยชิ้นนี้เสนอขั้นตอนวิธีการผสมรวมกันของข้อมูลที่สนับสนุนการตัดสินใจได้ดีขึ้น ขั้นตอนการทำงานประกอบไปด้วย 1) การตรวจจับใบหน้า จมูก และปากโดยใช้ Multi-task Cascaded Convolutional Networks (MTCNN) 2) การสกัดคุณลักษณะรูปใบหน้า รูปจมูก รูปปาก และ 3) การจำแนกประเภทใบหน้าจากการผสมรวมข้อมูลเข้าหลายค่าโดยใช้ CNN ข้อมูลที่ใช้ในการทดลองคือข้อมูล Face Mask Label Dataset (FMLD) และข้อมูล Andrewmvd Face Mask Detection Kaggle (AFMDK) ผลการทดลองโมเดลแบบการผสมรวมข้อมูลเข้าหลายค่า (Multiple Inputs Ensembling: MIE) ที่นำเสนอมีความถูกต้อง ค่าความแม่นยำ ค่าความครบถ้วน และค่า F1 ที่มีประสิทธิภาพสูงกว่าแบบข้อมูลเข้าเชิงเดี่ยวในทั้งสองชุดข้อมูล

คำสำคัญ — การจำแนกใบหน้า, โครงข่ายประสาทแบบคอนโวลูชัน, การสกัดคุณลักษณะ, การเรียนรู้เชิงลึก.

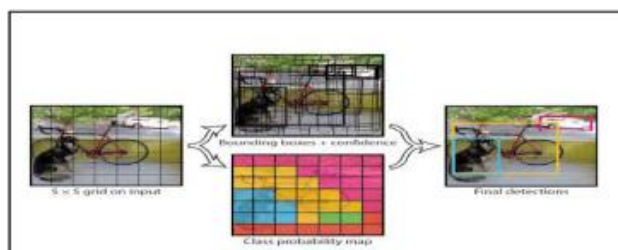
ABSTRACT — Object Detection can be used in many applications. Face mask classification for wearing a mask correctly, wearing a mask incorrectly, and not wearing a mask is a very important application for the recent situation. This research proposes the algorithm to ensemble multiple inputs for better decision making. The processes for this method are 1) face, nose, and mouth detections using Multi-task Cascaded Convolutional Networks (MTCNN), 2) feature extraction for face, nose, and mouth figures, and 3) face classification with multiple inputs ensembling using CNN. The datasets for this experiment are from Face Mask Label Dataset (FMLD) and Andrewmvd Face Mask Detection Kaggle (AFMDK). For both datasets, the experimental results show that the proposed model for Multiple Inputs Ensembling (MIE) has higher evaluation measurement of accuracy, precision, recall, and F1-score than those of the single input for both datasets.

Keywords — Face Classification, Convolutional Neural Network, Feature Extraction, Deep Learning.

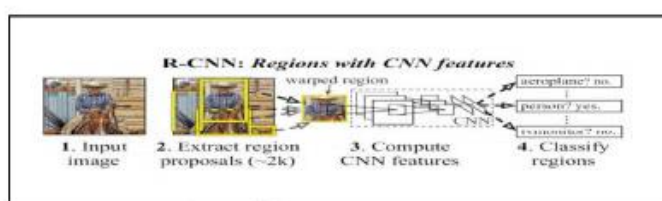
1. บทนำ

คอมพิวเตอร์วิทัศน์ (Computer Vision) เป็นสาขาหนึ่งของปัญญาประดิษฐ์ (Artificial Intelligent) มีเทคนิคการจดจำวัตถุ (Object Recognition) [1] เพื่อการจำแนกวัตถุและการจำกัดตำแหน่งของวัตถุ สามารถแบ่งออกเป็นสามกลุ่ม ได้แก่ 1) การจำแนกรูปภาพ (Image Classification) ว่าวัตถุในรูปนั้นเป็นวัตถุประเภทใด 2) การจำกัดตำแหน่งของวัตถุ (Object Localization) คือ การระบุตำแหน่งของวัตถุในรูปโดยสร้างกรอบรูป (Bounding Box: BB) แสดงบริเวณที่วัตถุอยู่และ 3) การตรวจจับวัตถุ (Object Detection) สามารถระบุวัตถุแต่ละวัตถุอยู่ที่ตำแหน่งใดของรูป และสามารถจำแนกประเภทวัตถุได้ การตรวจจับวัตถุจะใช้การสกัดคุณลักษณะ (Feature Extraction) เพื่อสกัดคุณลักษณะของรูป เช่น ขอบมุม และเส้นโค้ง เป็นต้น คุณลักษณะจะถูกนำไปใช้ในการจำแนกประเภทของวัตถุ การสกัดคุณลักษณะมีเทคนิคต่าง ๆ เช่น การตรวจจับขอบหรือการตรวจจับมุม เป็นต้น เนื่องจากวัตถุแต่ละตัวมีคุณลักษณะที่ต่างกันจึงต้องใช้เทคนิคการสกัดคุณลักษณะให้เหมาะสม การสกัดคุณลักษณะสามารถใช้การเรียนรู้เชิงลึก (Deep Learning) แทนเทคนิคต่าง ๆ ได้นอกจากนี้การจำกัดตำแหน่งของวัตถุสามารถใช้เทคนิคการเลื่อนช่อง (Sliding Window) ที่ใช้ช่องขนาดต่าง ๆ เพื่อจำกัดตำแหน่งของวัตถุ เทคนิคนี้ต้องใช้ทรัพยากรมากและส่งผลให้ใช้เวลานาน [2][3] ซึ่งการเรียนรู้เชิงลึกสามารถใช้แก้ปัญหาดังกล่าวได้

งานวิจัยนี้ได้ศึกษาและพัฒนาโมเดลเพื่อการรู้จำใบหน้าสวมหน้ากาก มีการสกัดคุณลักษณะและใช้ข้อมูลเข้าผสมมากกว่าหนึ่งค่าโดยใช้การเรียนรู้เชิงลึกหรือโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolutional Neural Network: CNN) เพื่อเพิ่มประสิทธิภาพให้สูงขึ้นในการจำแนกประเภท (Classification)



รูปที่ 1 แสดงการทำงานของ YOLO [7]



รูปที่ 2 แสดงขั้นตอนการทำงานของ R-CNN [8]

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1. การตรวจจับวัตถุ (Object Detection)

การตรวจจับวัตถุสามารถแบ่งตามแนวคิดที่ใช้ในการแก้ปัญหาเป็นสองประเภท ได้แก่ 1) การตรวจจับวัตถุที่ทำงานโดยใช้วิธีการถดถอย (Regression-based Object Detection) [4][5] และ 2) การตรวจจับวัตถุที่ทำงานตามพื้นที่ (Region-based Object Detection) [6]

การตรวจจับวัตถุที่ใช้แนวคิดการถดถอย เช่น โมเดล YOLO [7] การทำงานของ YOLO จะใช้ Convolutional Neural Network (CNN) 1 ตัว เพื่อสร้างกรอบรูปจำแนก (Predicted Bounding Box) และจำแนกประเภทของกรอบรูปไปพร้อม ๆ กัน การทำงานเริ่มต้นจากการแบ่งรูปออกเป็นตารางดังรูปที่ 1 หากช่องไหนมีศูนย์กลางของวัตถุอยู่ก็จะถือว่าช่องนั้นเป็นช่องหลักในการหาวัตถุนั้น โดยแต่ละช่องจะสร้างกรอบรูปออกมาจำนวนหนึ่ง ค่าความเชื่อมั่นของกรอบรูปแต่ละอันมีไว้เพื่อแสดงโอกาสที่กรอบรูปนั้นจะมีวัตถุอยู่ โดยในแต่ละช่องจะมีค่าความเป็นไปได้ที่บอกว่าวัตถุนั้นเป็นวัตถุประเภทใด

การตรวจจับโดยใช้แนวคิดเชิงพื้นที่ เช่น โมเดล R-CNN (Region-based CNN) [8] การทำงานของโมเดลนี้แสดงดังรูปที่ 2 เริ่มจากขั้นตอนที่เรียกว่า Region Proposal ซึ่งสร้างกรอบรูปในบริเวณที่สนใจ (Region of Interest: RoI) ซึ่งเป็นบริเวณที่คาดว่าจะมีวัตถุอยู่ หลังจากนั้นนำกรอบรูปไปสกัดคุณลักษณะและจำแนกประเภทของวัตถุ (Classification) ซึ่งจะใช้การถดถอยเพื่อทำให้กรอบรูปจากขั้นตอน Region Proposal พอดีกับขนาดของวัตถุโดยใช้โครงข่ายประสาทเทียม

2.2. โครงข่ายประสาทเทียม (Artificial Neural Network: ANN)

ANN เป็นเทคนิคที่เลียนแบบจากโครงข่ายประสาทของมนุษย์ โครงข่ายประสาทมีนิวรอน (Neuron) หรือโหนด (Node) โดยรับข้อมูลเข้า (Input) ที่เป็นข้อมูลจากชุดข้อมูล หรือข้อมูลจากนิวรอนในชั้นก่อนหน้า และส่งต่อให้นิวรอนในชั้นถัดไป โดยเส้นเชื่อมระหว่างนิวรอนแต่ละตัวจะมีค่า

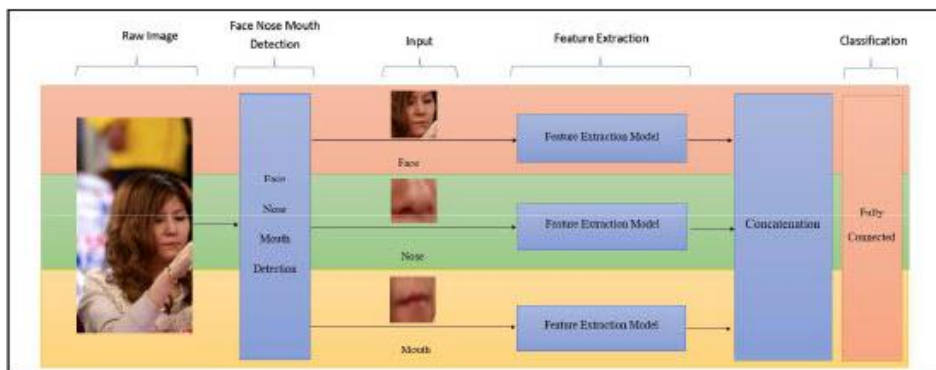
น้ำหนัก w มีเส้นเชื่อมนิวรอนนำข้อมูลเข้า x จำนวน n ตัว คูณกับค่าน้ำหนักจากนั้นนำผลลัพธ์ที่ได้มารวมกันบวกกับค่าความลำเอียง (Bias) w_0 หลังจากนั้นจะผ่าน Activation Function a ออกมาเป็นข้อมูลออก (Output) ของนิวรอนนั้น ซึ่งสามารถคำนวณได้ด้วยสมการที่ (1)

$$Output = a\left(\sum_{i=1}^n x_i * w_i\right) + w_0 \quad (1)$$

เมื่อข้อมูลเข้าถูกป้อนเข้าไปผ่านชั้นต่าง ๆ ในโครงข่ายประสาทเทียม จนถึงชั้นข้อมูลออก จะมีการคำนวณค่าความสูญเสีย (Loss) โดยคำนวณได้จากการเปรียบเทียบค่าข้อมูลออกจาก ANN กับค่าข้อมูลออกจริง ANN มีหลายชนิด เช่น Fully Connected Feedforward ANN เป็น ANN ที่นิวรอนเชื่อมถึงกันหมด โดยนิวรอนทุกตัวในแต่ละชั้นส่งข้อมูลให้นิวรอนในชั้นถัดไปทุกตัว และข้อมูลไหลไปทางเดียว สำหรับ Backpropagation ANN เป็น ANN ที่มีการป้อนค่าย้อนกลับพิจารณาจากอัตราการเปลี่ยนแปลง (gradient) ที่คำนวณได้จากค่าความสูญเสีย นำไปปรับค่าน้ำหนักเพื่อให้ข้อมูลออกมีความถูกต้องมากขึ้น

2.3. Convolutional Neural Networks (CNN)

CNN สามารถสกัดคุณลักษณะอัตโนมัติโดยใช้เทคนิค Backpropagation ที่ใช้ Gradient ที่คำนวณได้จากค่าความสูญเสียที่ได้จากการนำค่าที่จำแนกกับค่าที่เป็นจริงมาคำนวณ CNN ใช้ในการจำแนกประเภทรูปภาพ หลักการทำงานจะสร้างคัมพ้นที่การมองเห็นที่เรียกว่า Reception Field โดยพิกเซลของรูปภาพเป็นยูนิิตของ Reception Field การใช้งาน CNN สามารถใช้งานเพื่อลดรูปของรูปภาพให้มีคุณลักษณะลดลงให้เหลือเฉพาะคุณลักษณะที่สำคัญและง่ายต่อการเข้าใจ CNN หนึ่งตัวประกอบไปด้วย Convolutional Layer หลายชั้น ซึ่งแต่ละชั้นจะมี Filter หรือ Kernel ที่จะดำเนินการคูณเมตริกกับยูนิิตแต่ละตัวตามขนาดของ Filter ทำให้ผลลัพธ์ที่



รูปที่ 3 สถาปัตยกรรมการจำแนกด้วยการผสมรวมข้อมูลเข้าหลายค่า (Multiple Inputs Ensembling) โดยใช้ CNN

ได้เมื่อผ่าน CNN หนึ่งชั้นมีขนาดลดลงตามการตั้งค่าต่าง ๆ ได้แก่ ขนาดของ Filter ขนาดของ Stride ที่เป็นตัวกำหนดระยะในการเลื่อนของ Filter และค่า Dilate นอกจากนี้ CNN สามารถใช้ Padding เพื่อรักษารายละเอียดของข้อมูลเข้า และข้อมูลออกมีขนาดเท่ากันได้ เมื่อได้คุณลักษณะที่สำคัญแล้วจึงส่งให้ ANN ทำการจำแนกประเภทต่อไป

2.4. Multi-task Cascaded Convolutional Networks (MTCNN)

MTCNN เป็นโมเดลที่พัฒนาขึ้นเพื่อการตรวจจับใบหน้ามีผลลัพธ์ที่ได้เป็นกรอบรูปแสดงถึงใบหน้า ตำแหน่งของตา จมูก และปาก การทำงานจะใช้โมเดล 3 ตัวมาประกอบกัน ได้แก่ Proposal Network (P-Net), Refine Network (R-Net) และ Output Network (O-Net) P-Net จะทำหน้าที่ในการหาบริเวณที่น่าจะเป็นใบหน้าเบื้องต้น หลังจากนั้นจะส่งรูปไปให้ R-Net เพื่อจำกัดตำแหน่งให้ละเอียดขึ้น ขั้นตอนสุดท้ายรูปจะถูกส่งให้ O-Net เพื่อหาบริเวณตา จมูก และปากต่อไป โมเดลย่อยทั้ง 3 ตัวของ MTCNN มีการใช้งานแบบ Non-Maximum Suppression (NMS) เพื่อกำจัดกรอบรูปที่ซ้ำซ้อน โดยจะเลือกกำจัดจากรูปที่มีความมั่นใจต่ำกว่าก่อน [9]

2.5. งานวิจัยที่เกี่ยวข้อง

การตรวจจับการสวมหน้ากากในช่วงปัจจุบันมีการวิจัยและพัฒนาต่อเนื่องอย่างแพร่หลาย การตรวจจับใบหน้าโดยใช้แนวคิดเชิงถดถอย (Regression-based) ส่วนใหญ่ใช้โมเดล YOLO และ MobileNet ในการพัฒนาโมเดลตรวจจับใบหน้าสวมหน้ากากมีการใช้เทคนิคต่าง ๆ เช่น การใช้ Center loss เพื่อช่วยในการพัฒนาโมเดล [10][11] และการใช้ Spatial Separable Convolution และ Feature Enhancement Module (FEM) [12] เป็นต้น นอกจากนี้ในส่วนของงานการเรียนรู้ยังสามารถใช้เทคนิคการเรียนรู้ถ่ายโอน (Transfer Learning) ได้ด้วย สำหรับการทำงานของระบบสามารถทำงานได้ในลักษณะของเวลาจริง (Real Time) เช่นกัน สำหรับการตรวจจับใบหน้าเชิงพื้นที่ (Region-based) ใช้ Region Proposal หรือโมเดลการเรียนรู้เชิงลึกเพื่อหาใบหน้าจากรูป [6] [13] โดยมีโมเดลพื้นฐานในการจำแนกประเภท เช่น ResNet50, VGG16, MobileNetV2 และ Custom CNN ในส่วนของการตรวจจับบริเวณที่สนใจ (ROI Detection) มีการใช้โมเดลที่หลากหลาย เช่น MTCNN, SSD และ YOLOv4 เป็นต้น [16-19]

3. ขั้นตอนการดำเนินการวิจัย

ขั้นตอนการผสมรวมข้อมูลเข้าหลายค่า (Multiple Inputs Ensembling: MIE) ในการจำแนกใบหน้าโดยใช้ CNN นั้นจะใช้ข้อมูลเข้าเป็นรูปใบหน้า รูปจมูก และรูปปากมาผสมเข้าด้วยกันโดยมีแนวคิดคือต้องการเน้นบริเวณที่แสดงถึงคุณลักษณะที่แตกต่างกันอย่างชัดเจนของแต่ละประเภทข้อมูล ในการตรวจจับใบหน้าใสห้าหากมีประเด็นบริเวณจมูกและบริเวณปากดังนี้ เมื่อไม่ใสห้าหากจะเห็นจมูก และปากชัดเจน ในขณะที่ใสห้าหากไม่ถูกต้องอาจเห็นจมูกหรือปากแค่บางส่วน และในกรณีที่ใสห้าหากถูกต้องจะไม่เห็นทั้งจมูกและปาก ดังนั้นจึงได้มีการออกแบบโมเดลที่ใช้การตรวจจับใบหน้า จมูก และปากเป็นสถาปัตยกรรมข้อมูลเข้าแบบผสมรวมหลายค่า (MIE) โดยใช้ CNN เพื่อนำไปจำแนกการใสห้าหากแสดงดังรูปที่ 3 ขั้นตอนการดำเนินการเริ่มจากนำรูปต้นฉบับไปหาบริเวณใบหน้า จากนั้นจะหาบริเวณจมูก และหาบริเวณปาก หลังจากนั้นรูปทั้ง 3 รูป จะถูกส่งไปสกัดคุณลักษณะโดยใช้โมเดล CNN คุณลักษณะที่ได้ถูกรวมกันแล้วจะส่งไปที่ชั้น Fully Connected (FC) เพื่อทำการจำแนกประเภทโดยมีรายละเอียดดังต่อไปนี้

3.1. การตรวจจับใบหน้า จมูก และปาก (Face Nose Mouth Detection)

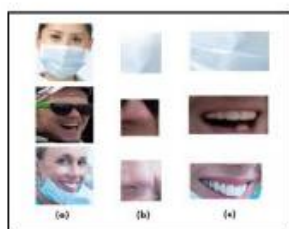
โมเดลที่ใช้ในการตรวจจับใบหน้า จมูก และปาก คือ Multi-task Cascaded Convolutional Networks (MTCNN) ตัวอย่างแสดงดังรูปที่ 4 บริเวณที่มีจุดสีน้ำเงินแสดงตำแหน่งของตาตำแหน่ง 2 ข้าง จุดสีแดงแสดงถึงตำแหน่งจมูก และจุดสีเขียวแสดงถึงตำแหน่งของขอบริมฝีปากทั้ง 2 ข้าง ในการวิจัยผู้วิจัยมีวิธีดำเนินการตั้งนี้รูปใบหน้าใช้ขนาด 224 x 224 รูปจมูกใช้ตำแหน่งจมูกที่เป็นจุดสีแดงขยายออกจากจุดศูนย์กลางให้มีขนาด 50 x 50 และรูปปากนำตำแหน่งจุดสีเขียว 2 จุดมาหาจุดกึ่งกลางแล้วขยายออกให้มีขนาดกว้างและยาวคือ 50 x 100 ผลลัพธ์ตัวอย่างการทำงานของรูปใบหน้า รูปจมูก และรูปปากแสดงได้ดังรูปที่ 5

3.2. การสกัดคุณลักษณะ (Feature Extraction)

การสกัดคุณลักษณะจะใช้โมเดล CNN Resnet152 จำนวน 3 ตัว โดยใช้ข้อมูลเข้ารูปใบหน้า รูปจมูก และรูปปากตามลำดับ ในการเตรียมข้อมูล



รูปที่ 4 ตัวอย่างจุดแสดงถึงตำแหน่งของตา ปาก และจมูกบนใบหน้า



รูปที่ 5 ตัวอย่าง (a) รูปใบหน้า (b) รูปจมูกและ (c) รูปปาก

ทุกชุดข้อมูลรูปจะปรับเป็นขนาด 224×224 หลังจากนั้นค่าแต่ละค่าในพิกเซลจะทำการ normalize ให้เป็นค่ามาตรฐานในช่วง -1 ถึง 1

3.3. การจำแนกประเภท (Classification)

การวิจัยใช้ Resnet152 ในการจำแนกประเภท ซึ่งการจำแนกประเภทของ Resnet จะใช้ชั้น Global Average Pooling (GAP) ตามด้วยชั้น Fully Connected (FC) การใช้ชั้น GAP ช่วยลดทรัพยากรที่ใช้งานลงเมื่อเทียบกับการใช้งาน FC หลาย ๆ ชั้น และแก้ปัญหา Overfitting ของ FC ได้ โดยผลลัพธ์ของ FC จะมีจำนวนของโหนดเท่ากับประเภทที่ต้องการจำแนก

4. การทดลอง

4.1. การจัดเตรียมข้อมูล

การทดลองได้ใช้ข้อมูลจำนวน 2 ชุด ได้แก่ 1) Face Mask Label Dataset (FMLD) [18] ซึ่งเป็นชุดข้อมูลที่เกิดจากการผสมกันระหว่างรูปจากชุดข้อมูล MAFA [19] 30,811 รูป และชุดข้อมูล Wider face [20] 11,123 รูป ทำให้ FMLD มีรูปทั้งหมด 41,937 รูป แต่ละรูปมีมากกว่าหนึ่งใบหน้า การตรวจจับใบหน้าจาก MTCNN ได้ค่าความถูกต้อง 59.75% โดยใช้ IOU มีค่ามากกว่าหรือเท่ากับ 0.5 ชุดข้อมูลมี 3 ประเภท (Class) ได้แก่ ประเภทใส่หน้ากากถูกต้อง 29,561 หน้า ประเภทไม่ใส่หน้ากาก 26,968 หน้า และประเภทใส่หน้ากากไม่ถูกต้อง 1,531 หน้า รวมทั้งหมด 58,060 หน้า การทดลองจะเลือกสุ่มมาจำนวน 5,000 หน้า ชุดข้อมูล FMLD 5K สร้างจากการสุ่มเลือกภาพจาก FMLD แบ่งออกเป็น 3 ประเภท คือ ประเภทใส่หน้ากากถูกต้อง (Class A) 2,500 หน้า ประเภทไม่ใส่หน้ากาก (Class B) 1,298 หน้า และประเภทใส่หน้ากากไม่ถูกต้อง (Class C) 1,202 หน้า ตัวอย่างรูปในชุดข้อมูล FMLD แสดงดังรูปที่ 6 สำหรับข้อมูลชุดที่ 2) Andrewmvd Face Mask Detection Kaggle (AFMDK) [21] มีรูปจำนวน 853 รูป การตรวจจับใบหน้าจาก MTCNN ได้ค่าความถูกต้อง 55.08% แต่ละรูปมีมากกว่าหนึ่งใบหน้า โดยใช้ IOU มีค่ามากกว่าหรือเท่ากับ 0.5 แบ่งเป็น 3 ประเภท คือ ประเภทใส่หน้ากากถูกต้อง (Class A) 3,232 หน้า ประเภทไม่ใส่หน้ากาก (Class B) 717 หน้า และประเภทใส่หน้ากากไม่ถูกต้อง (Class C) 123 หน้า รวมทั้งหมด 4,072 หน้า



รูปที่ 6 รูปตัวอย่างชุดข้อมูล FMLD (a) ใส่หน้ากากถูกต้อง (b) ไม่ใส่หน้ากาก และ (c) ใส่หน้ากากไม่ถูกต้อง

ข้อมูลทั้ง 2 ชุดถูกแบ่งออกเป็นอัตราส่วนสำหรับฝึกและทดสอบ 70:30 ใช้โมเดล Resnet152 ในการฝึกและทดสอบ

4.2. ผลการทดลอง

การทดลองเป็นการใช้ชุดข้อมูล FMLD 5K และ AFMDK โดยมีการทดลองและประเมินผลการจำแนก 2 รูปแบบ คือ รูปแบบที่หนึ่งเป็นการจำแนกเมื่อมีการแบ่งข้อมูลออกเป็น 2 ประเภท ดังผลในตารางที่ 1 และตารางที่ 2 รูปแบบที่สองเป็นการจำแนกเมื่อมีการแบ่งข้อมูลออกเป็น 3 ประเภท ดังผลในตารางที่ 3 และตารางที่ 4 รายละเอียดการทดลองของตารางที่ 1 ค่า Single Input แสดงถึงโมเดล Resnet152 ที่ใช้ข้อมูลเข้าเป็นรูปใบหน้าหรือรูปจมูกหรือรูปปากเพียงอย่างเดียวเท่านั้น สำหรับ Multiple Inputs คือ โมเดลที่ใช้ข้อมูลแต่ละรูปภาพรวมกัน เช่น face + nose + mouth คือ โมเดลที่ใช้รูปใบหน้า รูปจมูกและรูปปากรวมกัน เป็นต้น สำหรับการทดลองแบ่งข้อมูลเป็น 2 ประเภท คือ ประเภทที่ (Class 1) ใส่หน้ากากถูกต้อง กับ (Class 2) ประเภทไม่ใส่หน้ากากรวมทั้งประเภทใส่หน้ากากไม่ถูกต้อง การประเมินประสิทธิภาพการทดลองใช้ค่าเฉลี่ยการวัดของทุกประเภทในการนำเสนอ คือ ค่าความถูกต้อง (Accuracy), ค่าความแม่นยำ (Precision), ค่าความครบถ้วน (Recall), และค่า F1 ตัวอย่างการคำนวณแสดงดังตารางที่ 5 เช่น การหาค่าเฉลี่ยของความแม่นยำ ค่าความครบถ้วน และค่า F1 ของแต่ละประเภท ค่าความแม่นยำของ face + nose + mouth ของ Class 1 (ใส่หน้ากากถูกต้อง) มีค่า 91.27% และ Class 2 (ไม่ใส่หน้ากากและใส่หน้ากากไม่ถูกต้อง) ค่าความแม่นยำมีค่า 94.59% ทำให้ได้ค่าเฉลี่ยความแม่นยำเท่ากับ 92.93% เป็นต้น

ชุดข้อมูลที่ 1 คือชุดข้อมูล FMLD 5K (2 class) ผลการทดลองที่ใช้ Multiple Inputs Ensembling (MIE) แบบรูปใบหน้าร่วมกับรูปจมูกและรูปปากมีประสิทธิภาพดีกว่าแบบเดี่ยว (Single Input) เทียบกับที่มีรูปใบหน้าเพียงอย่างเดียว มีค่าความถูกต้อง, ค่าความแม่นยำ, ค่าความครบถ้วน, และค่า F1 อยู่ที่ 92.87%, 92.93%, 92.87%, และ 92.86% ตามลำดับ แสดงดังตารางที่ 1 ชุดข้อมูลที่ 2 AFMDK (2 class) จะเห็นได้ว่า MIE แบบรูปใบหน้าร่วมกับรูปจมูกและรูปปากมีค่าผลการทดลองที่สูงกว่าแบบใบหน้าอย่างเดียวเช่นกันโดยมีค่าความถูกต้อง ค่าความแม่นยำ ค่าความครบถ้วน และค่า F1 อยู่ที่ 97.14%, 95.40%, 95.54%, และ 95.57% ตามลำดับ แสดงดังตารางที่ 2

สำหรับการแบ่งข้อมูลเป็น 3 ประเภท ตารางที่ 3 แสดงชุดข้อมูล FMLD (3 class) แบบ MIE ของรูปใบหน้า รูปจมูก และรูปปากมีประสิทธิภาพสูงกว่าแบบรูปใบหน้าอย่างเดียวเช่นกัน โดยมีค่าความถูกต้อง ค่าความแม่นยำ ค่าความครบถ้วน และค่า F1 อยู่ที่ 85.13%, 83.11%, 82.71%, และ 82.41% ตามลำดับ ตารางที่ 4 แสดงชุดข้อมูล

ตารางที่ 1 ผลการทดลองชุดข้อมูล FMLD 5K (2 class) ของข้อมูลเข้ารูปใบหน้า รูปจมูก รูปปากแบบเดี่ยวและแบบรวม

Dataset	Evaluation Measurement (%)	Single Input			Multiple Inputs			
		face	nose	mouth	nose + mouth	face + nose	face + mouth	face + nose + mouth
FMLD 5K (2 class)	Accuracy	92.13	81.40	75.40	81.87	92.67	92.67	92.87
	Precision	92.22	84.33	79.09	84.25	92.74	92.74	92.93
	Recall	92.13	81.40	75.40	81.87	92.67	92.67	92.87
	F1-score	92.13	80.99	74.60	81.55	92.66	92.66	92.86

ตารางที่ 2 ผลการทดลองชุดข้อมูล AFMDK (2 class) ของข้อมูลเข้ารูปใบหน้า รูปจมูก รูปปากแบบเดี่ยวและแบบรวม

Dataset	Evaluation Measurement (%)	Single Input			Multiple Inputs			
		face	nose	mouth	nose + mouth	face + nose	face + mouth	face + nose + mouth
AFMDK (2 class)	Accuracy	96.97	92.47	91.24	92.96	97.14	96.97	97.14
	Precision	95.27	93.59	90.07	94.4	95.40	95.14	95.40
	Recall	95.13	82.09	80.86	83.03	95.54	95.28	95.54
	F1-score	95.20	86.36	84.41	87.30	95.47	95.21	95.47

ตารางที่ 3 ผลการทดลองชุดข้อมูล FMLD 5K (3 class) ของข้อมูลเข้ารูปใบหน้า รูปจมูก รูปปากแบบเดี่ยวและแบบรวม

Dataset	Evaluation Measurement (%)	Single Input			Multiple Inputs			
		face	nose	mouth	nose + mouth	face + nose	face + mouth	face + nose + mouth
FMLD 5K (3 class)	Accuracy	84.00	68.27	67.67	71.80	85.00	84.73	85.13
	Precision	81.99	63.47	66.59	69.94	82.94	82.75	83.11
	Recall	81.34	58.95	59.92	63.22	82.43	82.18	82.71
	F1-score	81.12	59.54	61.29	64.69	82.16	81.97	82.41

ตารางที่ 4 ผลการทดลองชุดข้อมูล AFMDK (3 class) ของข้อมูลเข้ารูปใบหน้า รูปจมูก รูปปากแบบเดี่ยวและแบบรวม

Dataset	Evaluation Measurement (%)	Single Input			Multiple Inputs			
		face	nose	mouth	nose + mouth	face + nose	face + mouth	face + nose + mouth
AFMDK (3 class)	Accuracy	96.40	91.41	90.43	91.57	96.48	96.24	96.40
	Precision	91.24	75.67	64.64	79.12	91.38	90.70	91.02
	Recall	83.03	61.24	55.33	62.38	83.06	80.94	82.08
	F1-score	86.01	65.80	58.20	67.63	86.10	84.28	85.23

ตารางที่ 5 ตัวอย่างผลการทดลองแยกตามประเภท FMLD 5K (2 class)

Type	Class	Precision (%)	Recall (%)	F1-score (%)
face	Class 1	90.31	94.40	92.31
	Class 2	94.13	89.87	91.95
	Average	92.22	92.13	92.13
face+ nose+	Class 1	91.27	94.80	93.00
	Class 2	94.59	90.93	92.73
	Average	92.93	92.87	92.86

AFMDK (3 class) MIE ของรูปใบหน้า และรูปจมูกมีประสิทธิภาพสูงกว่าแบบรูปใบหน้าเดี่ยว โดยมีค่าความถูกต้อง ค่าความแม่นยำ ค่าความครบถ้วน และค่า F1 อยู่ที่ 96.48%, 91.38%, 83.06%, และ 86.10% ตามลำดับ

5. สรุปและอภิปรายผล

งานวิจัยนี้ได้ศึกษาการพัฒนาโมเดลการผสมรวมข้อมูลเข้าหลายค่าโดยฝึกโมเดลการเรียนรู้เชิงลึกที่รับข้อมูลเข้าจำนวนมากว่าหนึ่งค่าการทดลองใช้โมเดล Resnet152 ในการทำงาน ผลการทดลองจากข้อมูลทั้ง

สองชุดมีประสิทธิภาพสูงขึ้นทั้งค่าความถูกต้อง ค่าความแม่นยำ ค่าความครบถ้วน และค่า F1 จากตัวอย่างชุดข้อมูล FMLD 5K (3 class) ในตารางที่ 3 ค่าความถูกต้องโมเดลที่นำเสนอ MIE ของรูปใบหน้า รูปจมูกและรูปปาก คือ 85.13% ในขณะที่ใช้ใบหน้าเพียงอย่างเดียวได้ค่าความถูกต้องเพียง 84.00% ค่าความแม่นยำ MIE ได้ 83.11% ในขณะที่ใช้ใบหน้าอย่างเดียวได้ 81.99% ค่าความครบถ้วน MIE ได้ 82.71% ในขณะที่ใช้ใบหน้าอย่างเดียว 81.34% และค่า F1 จาก MIE ได้ 82.41% ในขณะที่ใช้ใบหน้าอย่างเดียวได้น้อยกว่าคือ 81.12% ข้อสังเกตในชุดข้อมูล คือ AFMDK (2 class) ตารางที่ 2 การใช้ MIE เพียง 2 ค่าคือรูปใบหน้าและรูปจมูก และชุดข้อมูล AFMDK (3 class) ตารางที่ 4 MIE 2 ค่า คือรูปใบหน้าผสมรูปจมูกก็มีผลการทดลองที่สูงกว่าภาพใบหน้าแบบเดี่ยวได้ สาเหตุที่ทำให้ผลการทดลองมีประสิทธิภาพสูงขึ้น เนื่องจากการใช้รูปเดี่ยวที่เป็นรูปใบหน้าเพียงอย่างเดียว รูปจมูกเพียงอย่างเดียว หรือรูปปากเพียงอย่างเดียวได้ผลการทดลองที่ต่ำกว่า แต่เมื่อนำทั้งสามส่วนมารวมกันจะทำให้มีความสามารถในการแยกแยะมากยิ่งขึ้น หลักการทำงานดังกล่าวอาจนำไปประยุกต์ใช้ร่วมกับข้อมูลชนิดอื่น เช่น การตรวจจับการใส่หน้ากากของพนักงานในโรงงานอุตสาหกรรม หรือการสวมใส่หมวกนิรภัย เป็นต้น งานที่จะพัฒนาต่อไปในอนาคต คือ การพิจารณาลักษณะของข้อมูลเข้าเป็นรูปที่มีขนาดเล็กมาก หรือข้อมูลเข้าที่เป็นภาพที่ไม่ชัดเจน

เอกสารอ้างอิง

- [1] S. Hayat, S. Kun, Z. Tengtao, Y. Yu, T. Tu, and Y. Du, "A deep learning framework using convolutional neural network for multi-class object recognition," in 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), 2018, pp. 194–198.
- [2] N. O' Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. Velasco-Hernandez, L. Krpalkova, D. Riordan, and J. Welsh, "Deep learning vs. traditional computer vision," 10, 2019.
- [3] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212–3232, 2019.
- [4] M. R. Bhulayan, S. A. Khushbu, and M. S. Islam, "A deep learning based assistive system to classify covid-19 face mask for human safety with yolov3," in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1–5.
- [5] C. Tang, Y. Ling, X. Yang, W. Jin, and C. Zheng, "Multi-view object detection based on deep learning," Applied Sciences, vol. 8, p. 1423, 08, 2018.
- [6] J. Gathani and K. Shah, "Detecting masked faces using region-based convolutional neural network," in 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), 2020, pp. 156–161.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, realtime object detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
- [8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
- [9] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," CoRR, vol. abs/1604.02878, 2016.
- [10] X. Wang, Z. Chen, B. Wei, and M. Ling, "Application of pruning yolov4 with center loss in mask wearing recognition for gymnasiums and sports grounds of colleges and universities," in 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1373–1377.
- [11] G. Yang, W. Feng, J. Jin, Q. Lei, X. Li, G. Gui, and W. Wang, "Face mask recognition system with yolov5 based on image recognition," in 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1398–1404.
- [12] W. Han, Z. Huang, A. kuerban, M. Yan, and H. Fu, "A mask detection method for shoppers under the threat of covid-19 coronavirus," in 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), 2020, pp. 442–447.
- [13] J. Zhang, F. Han, Y. Chun, and W. Chen, "A novel detection framework about conditions of wearing face mask for helping control the spread of covid-19," IEEE Access, vol. 9, pp. 42975–42984, 2021.
- [14] I. Buciu, "Color quotient based mask detection," in 2020 International Symposium on Electronics and Telecommunications (ISETC), 2020, pp. 1–4.
- [15] J. R. V. Jeny, B. Shradha, B. Ashritha, D. S. Sai, and M. Naveen, "Deep learning framework for face mask detection," in 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 2021, pp. 1705–1712.
- [16] S. Abbasi, H. Abdi, and A. Ahmadi, "A face-mask detection approach based on yolo applied for a new collected dataset," in 2021 26th International Computer Conference, Computer Society of Iran (CSICC), 2021, pp. 1–6.
- [17] S. Sakshi, A. K. Gupta, S. Singh Yadav, and U. Kumar, "Face mask detection system using cnn," in 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021, pp. 212–216.
- [18] B. Batagelj, P. Peer, V. Štruc, and S. Dobrišek, "How to correctly detect face-masks for covid-19 from visual information?" Applied Sciences, vol. 11, no. 5, 2021.
- [19] S. Ge, J. Li, Q. Ye, and Z. Luo, "Detecting masked faces in the wild with lle-cnns," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 426–434.
- [20] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," CoRR, vol. abs/1511.06523, 2015.
- [21] J. leamsaard, S. N. Charoensook, and S. Yamman, "Deep learning-based face mask detection using yolov5," in 2021 9th International Electrical Engineering Congress (IEECON), 2021, pp. 428–431.

ประวัติผู้เขียน

ชื่อ สกุล	นายภีมกานต์ รุจิราพงศ์		
รหัสประจำตัวนักศึกษา	6310220004		
วุฒิการศึกษา			
วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา	
วิศวกรรมศาสตรบัณฑิต	มหาวิทยาลัยสงขลานครินทร์	2560	

การตีพิมพ์เผยแพร่ผลงาน

P. Rujirapong, W. Wettayaprasit, N. Wattanakitrunroj, “The Multiple Inputs Ensembling for Face Mask Classification Using CNN,” in The Fourteen National Conference on Information Technology (NCIT2022), 2020, pp.1-6